UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática

Contribuição dos Algoritmos de Difusão para o Desempenho dos Protocolos de Encaminhamento para Redes Ad Hoc Móveis

João Pedro Gomes Gouveia de Matos

MESTRADO EM ENGENHARIA INFORMÁTICA
Especialização em Arquitectura, Sistemas e Redes de Computadores

2009

UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática

Contribuição dos Algoritmos de Difusão para o Desempenho dos Protocolos de Encaminhamento para Redes Ad Hoc Móveis

João Pedro Gomes Gouveia de Matos

DISSERTAÇÃO

Projecto orientado pelo Prof. Doutor Hugo Alexandre Tavares Miranda

MESTRADO EM ENGENHARIA INFORMÁTICA
Especialização em Arquitectura, Sistemas e Redes de Computadores

2009

Agradecimentos

Ao Doutor Hugo Miranda que de forma amável e disponível acompanhou e orientou este estudo e cujas sugestões possibilitaram que eu encontrasse e percorresse o caminho que me permitiu a concretização de mais uma etapa da minha formação académica.

A todos os Professores do Mestrado em Engenharia Informática, especialização em Arquitectura de Sistemas e Redes de Computadores, pelas aprendizagens e reflexões que proporcionaram e que foram fundamentais para a realização da presente investigação.

Aos meus colegas do curso de Mestrado pelo companheirismo e empatia gerada, contribuindo para amenizar o esforço e os sacrifícios que a frequência de um curso exige.

À minha família pela força que me deu e que foi fundamental para chegar até aqui.

Aos meus amigos que, em momentos críticos da minha vida, me demonstraram que eu era capaz de levar a tarefa a bom porto.

Este trabalho foi parcialmente suportado pelas acções integradas luso-britânicas e pela FCT, através do projecto PTDC/EIA/71752/2006, REDICO: Dynamic Reconfiguration of Communication Protocols e pelo programa de financiamento plurianual.

Resumo

As redes ad hoc móveis (MANETs) são compostas exclusivamente pelos dispositivos dos participantes. A utilização destas redes revela-se de particular importância nas situações em que a instalação antecipada de infra-estruturas de suporte não é possível ou desejável.

Os algoritmos de difusão realizam a entrega de mensagens ao maior número possível de participantes na rede. A difusão é uma peça fundamental de muitos dos protocolos que se antecipa virem a ser usados nas MANETs. Dada a escassez de recursos que caracteriza estas redes, é da maior importância que a disseminação seja concretizada com o menor custo possível, tendo sido propostas alternativas ao dispendioso mas popular algoritmo de inundação.

Este trabalho avalia experimentalmente, através de simulações, o impacto de quatro algoritmos de difusão no desempenho do Ad hoc On Demand Distance Vector (AODV), um dos protocolos de encaminhamento mais citados para MANETs. Adicionalmente, é apresentada uma biblioteca para o sistema operativo Linux, que concretiza um dos algoritmos de difusão estudados.

Palavras-chave: Difusão, PAMPA, PAMPA2, Flooding, GOSSIP3(p, k, m), AODV.

Abstract

Mobile ad hoc networks (MANETs) are composed exclusively of the participants' devices. These networks are particularly relevant in scenarios where the deployment in advance of an infra-structure is not possible or desirable.

In MANETs, broadcast algorithms aim at delivering messages to the biggest number of participants. Broadcast operations are a fundamental building block for many protocols used in MANETs. Given that devices have limited resources, it is of utmost importance to reduce the resource consumption of each broadcast. Therefore, many alternatives to the costly, but popular, flooding algorithm, have been presented.

This document reports the experiments and evaluation, through simulations, of the performance of the Ad hoc On-demand Distance Vector (AODV) protocol with four distinct broadcast algorithms. This work also presents a library that implements one of the broadcasting algorithms evaluated in this document.

Keywords: Broadcast, PAMPA, PAMPA2, Flooding, GOSSIP3(p, k, m), AODV.

Conteúdo

Lista de Figuras

Listings

Capítulo 1

Introdução

A comunicação numa rede sem fios é, por norma, efectuada por uma infraestrutura de suporte, nomeadamente pontos de acesso ou satélites. Esta infraestrutura assegura as funções de gestão, como por exemplo a atribuição de endereços, bem como a prestação dos serviços básicos de comunicação, designadamente o escalonamento das transmissões dos dispositivos e o encaminhamento das mensagens. A título de exemplo, referem-se as redes GSM, UTMS e a maioria das redes baseadas nos protocolos da família IEEE 802.11, vulgo WiFi. Porém, pode acontecer que a instalação antecipada de uma infraestrutura não seja possível ou desejada, sendo certo que necessita da colocação e configuração das estações base que a concretizam. Neste tipo de situações incluem-se as operações de busca e salvamento e operações militares, sendo, nestas circunstâncias, as redes compostas exclusivamente pelos dispositivos participantes, os quais, duma maneira geral, asseguram os serviços disponibilizados por infraestruturas. A este tipo de redes dá-se o nome redes ad hoc móveis ou *MANETs*.

Os recursos energéticos, computacionais e comunicacionais dos dispositivos que compõem uma MANET são escassos, pelo que se pode afirmar, com toda a segurança, que a sua gestão eficiente é muito importante. As operações de rede dos dispositivos móveis, nomeadamente as de transmissão, assumem um carácter exigente em termos de consumo energético [5]. Da mesma forma, são necessários recursos energéticos em número suficiente para receber mensagens, bem como recursos computacionais para operações algorítmicas que permitam tratar uma mensagem recebida. Este trabalho considera que os dispositivos que compõem uma MANET são cooperantes e, consequentemente, disponibilizam os seus recursos para as operações de gestão de rede.

Os interfaces rádio dos dispositivos móveis têm um raio de alcance limitado, que se pode revelar insuficiente para cobrir todos os participantes da rede. Em consequência, os emissores podem não possuir dentro do seu raio de alcance, os respectivos participantes destinatários. Nesta circunstância, caberá aos participantes localizados entre um emissor e um destinatário concretizar a entrega, procedendo à retransmissão das mensagens. Serão estes que, adicionalmente, se responsabilizarão pelas operações de descoberta de

uma sequência de participantes, vulgarmente designada por rota, por forma a assegurar a comunicação entre os dois participantes em causa. O objectivo preferencial é obter a rota de menor custo, usualmente avaliada pelo número de participantes intermediários. Convém referir que se assume que os participantes da rede podem não ter conhecimento da sua localização, bem como da dos seus destinatários, facto que torna o problema consideravelmente mais complexo.

Para a descoberta da rota, usualmente utiliza-se um mecanismo que consiste na difusão de pequenas mensagens com indicação do destinatário. A forma de implementação mais comum consiste em obrigar todos os participantes a retransmitir todas as mensagens recebidas pela primeira vez. Esta operação é vulgarmente designada por inundação (do inglês *flooding*). Assim, se não ocorreram problemas de carácter físico, tais como interferências ou formação de partições[1], o participante destinatário, bem como todos os participantes da rede, vão receber a mensagem. Porém, é certo que se deve considerar que todos retransmitem sabendo que as retransmissões de apenas alguns conduzem ao mesmo objectivo.

Uma transmissão de um dispositivo, por exemplo como uma tentativa para contactar outro participante, cuja localização não lhe é conhecida, poderá gerar múltiplas retransmissões por parte dos participantes vizinhos, que por sua vez geram múltiplas retransmissões, continuando este processo a crescer exponencialmente. Assim, transmissões concorrentes, podem sobrecarregar o meio, promovendo a ocorrência de colisões, retransmissões redundantes e problemas de contenção. A colectividade destes três factores contribui para o aumento de consumo de recursos energéticos, computacionais e comunicacionais, bem como a perda de largura de banda útil, num processo conhecido como *broadcast storm* [13]. O uso do flooding, ao prever numerosas retransmissões desnecessárias pela rede, pode causar este fenómeno.

O flooding é um algoritmo de difusão utilizado em muitos protocolos para MANETs (por exemplo, [8, 11]). Recentemente, têm surgido novas ideias para algoritmos de difusão, tendo em vista a redução da sobrecarga da rede com retransmissões de mensagens, melhorando, consequentemente, o desempenho dos protocolos para MANETs e a gestão de recursos. Porém, quando aplicados em substituição do flooding, os resultados obtidos podem não ter a qualidade pretendida. Este trabalho analisa, o caso particular da aplicação de um algoritmo de difusão a um dos protocolos mais populares para redes ad hoc. Os resultados das simulações mostram uma redução da carga da rede, bem como uma melhoria significativa do desempenho do protocolo de encaminhamento, designadamente a nível de diminuição de latência, número de saltos nas rotas e taxa de entrega. Comparou-se ainda este desempenho com o resultante da aplicação de outros algoritmos ao protocolo, bem como com a sua versão original, nas mesmas condições. Finalmente, analisou-se

[1] Uma partição é definida pelo conjunto de dispositivos que se encontram no raio de transmissão de, pelo menos, um outro membro dessa partição.

ainda a implementação concretizada do algoritmo.

1.1 Contribuições

Este trabalho apresenta uma biblioteca, para o sistema operativo Linux, que concretiza o algoritmo de difusão *Power Aware Message Propagation Algorithm* (PAMPA).

Propõe uma nova versão do PAMPA, denominada por PAMPA2, para casos em que se pretende uma maior perenidade das rotas no AODV, quando se verifica muito movimento por parte dos participantes.

Apresenta uma versão optimizada do AODV, no sentido em que foi provado que, a substituição do flooding pelo PAMPA ou pelo PAMPA2, na difusão dos pedidos de rota, reflecte-se positivamente no desempenho do protocolo, gestão de recursos dos dispositivos e atenuação de problemas como *Broadcast Storm*.

1.2 Estrutura do documento

O relatório está organizado da seguinte forma: O capítulo 2 enquadra o trabalho relacionado, onde são apresentados alguns algoritmos de difusão, e um protocolo de encaminhamento.

No capítulo 3 são discutidas alternativas à utilização de algoritmos de difusão baseados em flooding nos protocolos de encaminhamento. É, também, apresentado um novo algoritmo de difusão.

No capítulo 4 são avaliados e comparados resultados de simulações dos diferentes algoritmos de difusão apresentados nos Capítulos 2 e 3, quando aplicados a um protocolo de encaminhamento, sob as mesmas condições.

No capítulo 5 é enquadrada uma implementação do PAMPA em ambiente Linux, que foi concretizada para placas de rede IEEE 802.11 (WiFi).

No capítulo 6 são apresentadas as conclusões mais importantes tiradas do trabalho realizado.

1.3 Publicações

O artigo ”Algoritmos de Difusão para Protocolos de Encaminhamento em Redes Ad Hoc sem Fios”que resume este trabalho, foi aceite para publicação no Simpósio de Informática (INFORUM 2009).

Capítulo 2

Trabalho relacionado

2.1 Algoritmos de Difusão

O objectivo dos algoritmos de difusão é a entrega, em melhor esforço, das mensagens disseminadas ao maior número possível de participantes. Pretende-se que o consumo de recursos seja o mais reduzido possível e que o desempenho do protocolo ao qual o algoritmo é aplicado seja o melhor possível. Para satisfazer estas exigências poder-se-á tentar atenuar o problema de *Broadcast Storm*, impedindo que alguns participantes retransmitam. Existem diversas formas de concretizar uma redução de número de retransmissões, como por exemplo, implementar um algoritmo de difusão, com contagem ou decisão probabilista.

2.1.1 Algoritmos Baseados em Contagem

Quando um participante recebe uma mensagem, pode não conseguir a sua retransmissão imediata. De facto, o intervalo de tempo compreendido entre a recepção da mensagem e a sua retransmissão pode estar dependente de factores como o tempo de computação ou a fila de espera. Durante esse tempo, é possível que o dispositivo receba várias retransmissões da mensagem que tenciona retransmitir. Assim, quando a mesma mensagem é recebida várias vezes, torna-se óbvio que a sua retransmissão é desnecessária, por a respectiva disseminação já ter sido assegurada pelos participantes vizinhos. Um algoritmo baseado em contagem assenta sobre este princípio.

É definida uma variável cujo valor indica o número de cópias que têm de ser recebidas para que a retransmissão seja cancelada. Para cada mensagem em espera, é criado um contador que incrementa sempre que uma cópia for recebida. Desta forma, se o valor do contador não ultrapassar o da variável, a mensagem é retransmitida, caso contrário, é descartada.

CB

Foi proposto em [13] um algoritmo de difusão baseado em contagem, denominado *Counter-based Broadcasting*, sendo que, os autores referem-se a ele como *CB*. Contrariamente ao flooding, em que os participantes retransmitem todas as mensagens recebidas pela primeira vez, o funcionamento deste algoritmo passa pela colocação em fila de espera das mensagens recebidas pela primeira vez por um intervalo de tempo escolhido aleatoriamente, sendo certo que tem de ser sempre inferior a um valor máximo pré-definido. Para isso, dá-se a inicialização de um contador com o valor 1, a ser incrementado cada vez que uma cópia da mensagem é recebida, sendo previamente atribuído um valor a uma variável de *treshold*, designada por *CH*. Assim, para cada mensagem em espera, se o contador atingir o valor atribuído a *CH*, a mensagem é descartada, não havendo lugar à retransmissão agendada. Se isto não se verificar, quando o tempo de espera imposto expirar, a mensagem é retransmitida.

Para a transmissão ou retransmissão de cada dispositivo, deverão ocorrer muito menos retransmissões dos respectivos participantes vizinhos, já que, algumas delas serão canceladas. Poder-se-á, assim, prever uma redução significativa do tráfego desnecessário comparativamente ao flooding. No entanto, é preciso ter em conta que o algoritmo não tem qualquer critério na selecção dos dispositivos para a retransmissão, visto que o tempo de espera é gerado aleatoriamente. Na verdade, os cancelamentos estão previstos para os dispositivos que impõem um tempo de espera maior às mensagens, sendo que estes podem ser os mais indicados para retransmitir, devido à sua localização. Este facto será constatado mais à frente neste trabalho. Esta desvantagem é comum a muitos algoritmos existentes actualmente.

2.1.2 Algoritmos Probabilistas

Para reduzir retransmissões de mensagens, poder-se-á utilizar um algoritmo probabilista. Estes algoritmos de difusão condicionam probabilisticamente a decisão dos participantes de retransmitir, convencionando um parâmetro de configuração p que representa a probabilidade de um participante retransmitir uma mensagem recebida pela primeira vez. De notar que com $p = 1$, estes algoritmos comportam-se de forma equivalente ao algoritmo do flooding. Se for atribuído um valor mediano a p, prevê-se que alguns participantes não retransmitam.

Os algoritmos baseados exclusivamente nesta propriedade não são facilmente adaptáveis a diferentes condições de rede. Aplicando a p um valor moderado, a possibilidade de, numa região onde o número de participantes é reduzido e todos decidem não retransmitir, é bastante real, resultando na terminação precoce da difusão. Por outro lado, atribuir valores mais elevados a p pode resultar num número excessivo de retransmissões, em zonas onde a concentração de dispositivos é mais elevada.

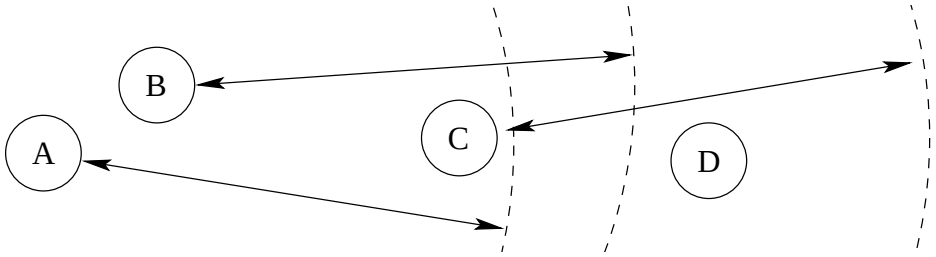


Figura 2.1: Raio de transmissão de 3 dispositivos

GOSSIP3(p, k, m)

O algoritmo GOSSIP3(p, k, m) [6] é um algoritmo probabilista regulado por três parâmetros p, k e m. O valor atribuído a p define a probabilidade de um participante retransmitir uma mensagem, mas o algoritmo não assenta exclusivamente nesta propriedade. Por forma a adaptar o comportamento a diferentes condições de rede, o GOSSIP3(p, k, m) propõe reter uma mensagem numa fila, durante um intervalo de tempo pré-definido, sempre que o dispositivo decide não retransmitir em resultado da função probabilista. A mensagem é descartada após o termo do tempo, caso se considere que foram recebidas cópias suficientes para assegurar a continuidade da propagação, ou é retransmitida caso contrário. O número de cópias considerado suficiente é definido pelo parâmetro m. Esta propriedade permite atribuir valores moderados ao parâmetro p reduzindo o risco de termo precoce da disseminação em determinadas regiões da rede. O GOSSIP3(p, k, m) é, portanto, um algoritmo baseado em contagem, além de probabilista.

A variável k assegura a continuação da difusão nos primeiros momentos, que podem ser vistos como uma janela de vulnerabilidade ao termo precoce, dado o reduzido número de dispositivos que receberam a mensagem até aquele momento. Os participantes que recebem a mensagem nos primeiros k saltos da difusão vão retransmiti-la com probabilidade 1, ignorando portanto, a função probabilista.

Assim sendo, as três variáveis p, k e m, determinam a instanciação do algoritmo, que é identificada pela expressão GOSSIP3(p, k, m). No entanto, a selecção dos dispositivos que retransmitem é aleatória, apesar de assegurar um número mínimo de retransmissões.

O impacto da ausência de qualquer heurística para a selecção dos dispositivos na eficiência dos algoritmos de difusão, está ilustrado na Figura 2.1. O dispositivo A retransmite uma mensagem, sendo entregue aos dispositivos B e C. Apesar de ambos apresentarem a mesma probabilidade de retransmissão, estas vão apresentar impactos diferentes. A retransmissão de B é sempre desnecessária por não contribuir com a entrega da mensagem a qualquer outro dispositivo. Acresce que, se C não retransmitir, a mensagem não será entregue ao dispositivo D. Ou seja, a propagação da mensagem não é garantida caso seja atribuído o valor 1 a m. Mesmo para valores de m superiores a 1, a propagação da mensagem não é garantida, assumindo a existência de outros dispositivos no raio de transmissão do emissor. Já foram demonstrados problemas similares em outros

algoritmos que concretizam a selecção probabilística dos dispositivos [10].

2.1.3 Algoritmos Baseados em Distância

O exemplo da Figura 2.1 sugere que os melhores candidatos à retransmissão são os dispositivos que se encontram mais afastados do emissor. Para valores reduzidos de distância entre um emissor e um participante receptor, a cobertura adicional proporcionada pela respectiva retransmissão é pequena. Portanto, poder-se-á afirmar que quanto maior for a distância entre o participante que emite e o participante que retransmite, maior será a cobertura adicional. Se este princípio for adoptado na decisão de retransmissão de cada dispositivo de uma MANET, a selecção dos dispositivos que retransmitem deixa de ser aleatória, como se verifica nos algoritmos probabilistas. Quando um dispositivo recebe uma mensagem, determina a distância a que se encontra do emissor e pode decidir não a retransmitir, se for inferior a um valor pré-definido, já que a sua cobertura adicional é considerada mínima. Um dispositivo pode estimar a distância a que se encontra do emissor, através da intensidade da força de sinal de recepção da mensagem, sabendo que o seu valor decai, de forma quadrática, com o aumento da distância entre os dois dispositivos.

PAMPA

O PAMPA (Power-Aware Message Propagation Algorithm) [9] é um algoritmo de difusão baseado em distância. Substitui a aleatoriedade da decisão de retransmitir dos dispositivos, pela distância entre eles, estimada através da intensidade da força de sinal da recepção de mensagens. A ideia passa por evitar que os dispositivos mais próximos do emissor retransmitam, deixando a disseminação para os mais afastados.

Para evitar operações de coordenação, cada dispositivo, ao receber uma mensagem, adia a retransmissão durante um intervalo de tempo calculado, multiplicando o valor da intensidade da força de sinal com que recebeu a mensagem, por uma constante pré-definida. De notar que o valor desta constante tem que ser igual para todos os dispositivos.

Assim, sabendo que o valor da intensidade da força de sinal decai com o aumento da distância, quanto mais longe um dispositivo se encontrar do emissor, mais depressa irá retransmitir.

Pretende-se que os participantes cancelem a retransmissão da mensagem quando são recebidas cópias de outros dispositivos, donde se conclui que este algoritmo também é baseado em contagem. É definida uma variável de *treshold* indicando o número máximo de cópias que têm de ser recebidas para que uma retransmissão seja cancelada. Durante o tempo de espera calculado, a mensagem é colocada numa fila para ser retransmitida quando o tempo expirar. Porém, a mensagem será descartada se forem recebidas cópias suficientes para concluir que a disseminação foi assegurada por participantes vizinhos geograficamente melhor posicionados. Isto é, dentro do raio de alcance de cada emissor, o

PAMPA vai escolher os participantes mais afastados, seleccionando, para cada mensagem difundida, aqueles que cuja localização proporciona uma maior cobertura.

Sabendo que para cada retransmissão de uma mensagem, o PAMPA selecciona os participantes em melhores condições geográficas para proporcionar maior cobertura, isto é, proporcionam mais entregas, intuitivamente se espera que este algoritmo não apresente os problemas relacionados com a falta de adaptação a diferentes condições de rede, verificados em muitos algoritmos probabilistas. Ou seja, o PAMPA tem este desempenho para regiões com elevada concentração de dispositivos e para regiões com poucos dispositivos.

Se for atribuído o valor 1 à variável de *treshold* do PAMPA, cada retransmissão obriga, em princípio, o cancelamento das retransmissões agendadas de todos os dispositivos localizados dentro da área da intersecção entre o raio de alcance do retransmissor e o raio de alcance do participante que lhe entregou a mensagem. Assim, prevê-se uma utilização de retransmissões bastante menor, relativamente a outros algoritmos de difusão, nomeadamente o flooding e o GOSSIP3(p, k, m).

2.2 Protocolos de Encaminhamento

2.2.1 AODV

O Ad hoc On-demand Distance Vector (AODV) [11] é um protocolo de encaminhamento para redes ad hoc móveis, no qual todos os participantes podem assumir o papel de encaminhador. Em cada dispositivo, a tabela de encaminhamento armazena, para cada destinatário conhecido, aquele a quem devem ser entregues as mensagens, isto é, o próximo salto.

O AODV é um protocolo reactivo, ou seja, os dispositivos participantes iniciam as operações de descoberta de rota sempre que necessitam (e apenas nestes casos) de uma rota para um dispositivo que não consta na sua tabela. Nestes casos, em que o participante emissor não possui qualquer informação sobre o destinatário na sua tabela de encaminhamento, transmite uma mensagem de pedido de rota (*route request*). Na sua versão original, o *route request* é difundido por um algoritmo de flooding. As tabelas de encaminhamento dos dispositivos participantes, são povoadas durante as operações de descoberta de rota.

A mensagem de *route request* dispõe, entre outros atributos, de campos para os endereços do emissor e do destinatário, bem como um contador destinado a apurar o número de saltos (incrementado a cada retransmissão). Existem ainda mais três campos (número de sequência do emissor, número de sequência da última rota conhecida para o destinatário e identificador de difusão) que permitem evitar a utilização de rotas desactualizadas. De facto, possuindo os dispositivos um contador para o identificador de broadcast, incrementado sempre que difundem uma mensagem, é sempre possível identificar duplicados de pedidos de rota para cada mensagem, através do par <endereço do emissor,identificador

de broadcast>.

Na prática, sempre que um participante recebe pela primeira vez uma mensagem de *route request*, o primeiro passo é verificar se existe, na sua tabela de encaminhamento, uma rota para o destinatário. Em caso afirmativo, dirigirá ao emissor uma mensagem de resposta (*route reply*). Só no caso contrário poderá ocorrer a retransmissão da mensagem. Da mesma forma, as mensagens de resposta são produzidas pelo destinatário sempre que recebe um *route request*. Na medida em que assumem a bidireccionalidade do canal de comunicação e independentemente do modo como são gerados, os *route reply* são enviados ponto-a-ponto, percorrendo a sequência inversa dos dispositivos que retransmitiram o *route request*. Desta forma, evita-se o recurso a mais uma dispendiosa operação de flooding.

De salientar que as rotas são criadas no sentido inverso ao da propagação das mensagens de *route request* e *route reply*. De facto, ao receber de um dispositivo A uma mensagem de pedido de rota produzida por um dispositivo B na sua tabela de encaminhamento, o dispositivo regista que as mensagens com destino a B devem ser entregues ao dispositivo A. Da mesma forma, o dispositivo que recebe de A um *route reply* para o dispositivo B regista na sua tabela de encaminhamento que A é o próximo salto das mensagens destinadas a B.

Refira-se que o *route reply* contém um campo para o número de saltos, pelo que um participante, ao receber vários *route replies* para o mesmo destino, pode basear-se neste campo para seleccionar a rota mais indicada para dar início à transmissão de dados.

Note-se que a mobilidade dos participantes pode invalidar rotas adquiridas no passado. Efectivamente, a deslocação dum participante para fora do alcance dos restantes que compõem a rota, é suficiente para quebrar a ligação. São os dispositivos intermédios, incapazes de entregar uma mensagem de dados ao dispositivo que consta na sua tabela de encaminhamento, que geram as mensagens de erro na rota (*route error*). Assim sendo, na grande maioria dos casos, quando o emissor de dados recebe uma destas mensagens, reage com difusão de um novo pedido de rota. Alguns participantes intermédios contêm na sua tabela de encaminhamento a rota para o destinatário, utilizada até ao momento, que se tornou inválida. Nestes casos, há interesse em não responder com um *route reply*.

Cada vez que um participante recebe um pedido de rota e constata que tem a entrada para o destinatário pretendido na sua tabela de encaminhamento, verifica se o número de sequência do destinatário que possui é inferior ao que está na rota. Se, de facto for inferior, o participante *não vai* gerar um *route reply*, mas sim dar continuidade à disseminação. Mas se o número de sequência que possui for igual ou superior ao do pedido de rota, vai responder. Quanto ao número de sequência do emissor, é utilizado para manter actual a informação sobre a rota inversa.

É, assim, previsível que uma mensagem de pedido de rota continue a ser difundida pela rede, mesmo depois do participante emissor ter obtido resposta. Este problema

assume visibilidade quando os participantes, emissor e destinatário (ou um participante que possua o próximo salto da rota em causa), se encontram próximos numa rede muito grande. Tendo em vista este problema, o AODV difunde a mensagem de pedido de rota através de um *expanding-ring search*, com o objectivo de limitar inicialmente o número de saltos do *route request* para que ela seja retransmitida apenas pelos participantes cuja localização está próxima do emissor. Caso não haja resposta, decorrido algum tempo, o processo é repetido com um número de saltos suficientemente maior para que seja recebida por todos os dispositivos da rede.

2.3 Sumário

A entrega de mensagens ao maior número possível de participantes numa MANET, é o principal objectivo dos algoritmos de difusão. Os algoritmos baseados em inundação impõem um custo muito elevado a nível de retransmissões, contribuindo directamente para o agravamento do problema de *Broadcast Storm*, aumentando o consumo de recursos dos dispositivos da rede, que se pretende ser o mais reduzido possível.

Foram estudados vários algoritmos de difusão que propõem a diminuição do número de dispositivos a retransmitir nas difusões de mensagens, como uma solução para reduzir o problema de *Broadcast Storm* e obter a consequente diminuição de consumo de recursos.

O *CB* impõe um atraso aleatório nas retransmissões dos dispositivos, obrigando alguns a cancelarem as suas após terem recebido um determinado número de retransmissões de participantes vizinhos. Porém, este algoritmo possibilita que os dispositivos mais indicados para retransmitir, cancelem as suas retransmissões.

O GOSSIP3(p, k, m) proporciona uma redução significativa do número de retransmissões, condicionando, probabilisticamente, a decisão de retransmitir dos dispositivos. Contudo, é esta a característica que impõe problemas de adaptação a diferentes condições de rede, ao algoritmo.

O PAMPA propõe impedir que os dispositivos mais próximos do emissor retransmitam. Cada dispositivo calcula o atraso a ser imposto à retransmissão de uma mensagem recebida, multiplicando o valor da intensidade da força de sinal com que a recebeu, por uma constante pré-definida, obrigando assim, os mais afastados a retransmitir mais depressa. Os participantes mais próximos do emissor cancelam as retransmissões, ao receberem cópias provenientes dos retransmissores mais afastados. Prevê-se que o PAMPA proporcione uma maior redução do número de retransmissões, relativamente ao GOSSIP3(p, k, m) e que não apresente problemas na adaptação a diferentes condições de rede.

O AODV é um protocolo de encaminhamento, que permite a comunicação entre quaisquer dois dispositivos de uma MANET. Para a descoberta de uma rota entre dois partici-

pantes, é difundida uma pequena mensagem de descoberta de rota, que se pretende que seja entregue ao maior número possível de participantes da rede, recorrendo ao flooding.

Capítulo 3

Alternativas à inundação no AODV

A difusão de mensagens é uma operação fundamental no AODV, pois permite a descoberta de rotas entre dois dispositivos da rede, garantindo a comunicação entre eles.

A quebra de rotas é um problema bastante real nos protocolos de encaminhamento para redes ad hoc móveis. É necessário ter em conta que os dispositivos de uma MANET podem deslocar-se livremente e que o alcance de transmissão de cada um dificilmente é elevado. Portanto, um ligeiro deslocamento de um dispositivo, pode ser suficiente para sair do alcance de alguns dos seus vizinhos e, consequentemente, quebrar uma ou mais rotas em que estava inserido. No AODV, uma quebra de rota leva a que o participante emissor difunda um novo pedido de rota. Se for verificado muito movimento numa MANET, é previsível que a quantidade de pedidos de rota suba consideravelmente, o que constitui um problema na medida em que o flooding é uma operação de difusão bastante dispendiosa, quer em termos de recursos dos dispositivos, quer de largura de banda, pelo que é da maior importância reduzir o custo das operações de difusão.

Contudo, a utilização do flooding na difusão dos pedidos de rota no AODV apresenta uma vantagem importante, já que uma operação de flooding permite descobrir todas as rotas existentes entre quaisquer dois dispositivos, uma vez que todos os participantes retransmitem o pedido de rota. Desta forma, um participante emissor pode escolher, entre todas as rotas que recebeu, a que considerar mais favorável, de acordo com o número de saltos e o congestionamento. Porém, um tão grande número de combinações é claramente desnecessário, pois muitas apresentarão custos muito elevados. O AODV impõe a cada dispositivo uma única retransmissão da mensagem, evitando por isso, a descoberta dessas rotas.

Com o modelo base de propagação do AODV, cada pedido de rota define um conjunto de rotas, todas elas disjuntas nos dispositivos intermédios e que incluirá sempre a rota mais favorável. No entanto, este sistema pode não proporcionar os resultados pretendidos, devido às perturbações introduzidas, ao nível de ligação de dados, pelo elevado número de transmissões concorrentes impostas pelo flooding. De facto, dependendo da política de controlo de acesso ao meio utilizada, o elevado número de transmissões concorrentes

pode gerar colisões e, consequentemente, algumas das melhores rotas podem não ser descobertas. Tipicamente é aplicado um atraso aleatório à retransmissão das mensagens, de modo a atenuar o problema da ocorrência de colisões, embora possa, igualmente, prejudicar as melhores rotas.

3.1 Aplicação de outros algoritmos no AODV

O AODV recorre ao flooding para difundir *route requests*, o que torna as operações de descoberta de rota muito dispendiosas. A substituição do flooding, por outro algoritmo de difusão de menor custo, no AODV, pode tornar as operações de descoberta de rota menos dispendiosas.

Os algoritmos de difusão alternativos contrastam com o flooding em muitos aspectos. A redução do número de dispositivos a retransmitir pedidos de rota, com base na selecção imposta pelo algoritmo, pode levar a que não sejam encontradas as melhores rotas, já que os dispositivos que a compõem podem não ser seleccionados. Por outro lado, a redução do número de retransmissões imposta, reduz fortemente o congestionamento, beneficiando o desempenho do protocolo, com tempos de descoberta de rota significativamente mais reduzidos e aumentando a largura de banda. É, portanto, pertinente comparar a viabilidade da aplicação de algoritmos de difusão alternativos ao flooding, ao AODV.

Uma primeira aproximação foi apresentada em [6]. Os autores comparam o desempenho do AODV com o flooding e com o GOSSIP3(p, k, m). Os estudos dos autores afirmam que a instância GOSSIP3$(0.65, 1, 1)$ proporciona um desempenho mais vantajoso para o AODV, nomeadamente a nível de taxa de entrega e de latência, para um conjunto de topologias genéricas. Os resultados da comparação sugerem que a substituição do flooding pelo GOSSIP3$(0.65, 1, 1)$ é benéfica [6], reduzindo o número de retransmissões de pedidos de rota em 35%. Esta redução diminuiu o número de colisões e, consequentemente, aliviou o problema de *Broadcast Storm*, o que teve impacto directo no desempenho do AODV, reduzindo consideravelmente a latência a nível de procura de rotas e a nível de entrega de mensagens de dados e aumentando a taxa de entrega das mesmas. Os autores de [6] denominaram a aplicação do GOSSIP3(p, k, m) ao AODV, por AODV+G.

No entanto, foi observado que as rotas obtidas são, entre 10% e 15%, mais longas do que as rotas obtidas pelo flooding. O comprimento das rotas obtidas pelo GOSSIP3(p, k, m) tende a aproximar-se do comprimento das rotas obtidas pelo flooding, à medida que se aumenta o valor do parâmetro de probabilidade de retransmissão [6]. O aumento da quantidade de dispositivos que retransmitem a mensagem, devido ao aumento do valor do parâmetro p, implica um acréscimo na probabilidade das retransmissões serem concretizadas pelos dispositivos que seriam seleccionados usando o flooding.

3.2 Aplicação do PAMPA ao AODV

Ao contrário do GOSSIP3(p, k, m), o PAMPA impõe uma política, não aleatória, de selecção dos dispositivos que retransmitem cada mensagem, bem como um atraso adicional à propagação. Aplicando o PAMPA ao AODV, dentro do raio de transmissão de cada emissor, são escolhidos os participantes mais afastados para assegurar a disseminação das mensagens de pedido de rota.

Quando um participante recebe um *route request*, verifica se dispõe de uma rota para o destinatário na sua tabela de encaminhamento ou se a mensagem já foi recebida. Em ambos os casos, o protocolo AODV original é respeitado, respondendo à mensagem com um *route reply* ou ignorando-a. É aplicada a política de retransmissão do PAMPA à primeira cópia de *route request* para o qual não se dispõe de rota na tabela de encaminhamento. O mesmo será dizer que, a mensagem é colocada numa fila de espera durante o intervalo de tempo calculado em função da intensidade da força de sinal com que recebeu o *route request*. Se durante esse tempo o participante receber uma cópia da mensagem em causa, vai removê-la da fila, bem como descartar a cópia que recebeu. Caso contrário, a mensagem é retransmitida quando o temporizador expirar. É, portanto, atribuído o valor 1 à variável de *treshold* do PAMPA. À aplicação do PAMPA ao AODV, demos o nome de AODV+P.

Espera-se que, o AODV+P apresente um menor número de retransmissões de *route requests* do que o AODV clássico e o AODV+G, pelo que é previsível que o desempenho do AODV+P, a nível de taxa de entrega de mensagens de dados, seja superior ao desempenho do AODV clássico bem como ao do AODV+G, devido à diminuição da ocorrência de *Broadcast Storms*. Acresce que, o AODV+P deverá apresentar, maioritariamente, rotas mais curtas do que as outras versões do AODV. Isto porque, o PAMPA deverá seleccionar os dispositivos localizados mais longe, dentro do raio de alcance de cada transmissor, para retransmitir os pedidos de rota, levando a que as rotas sejam descobertas com um menor número de saltos.

Com o propósito de visualizar o comportamento do AODV+P, foram realizadas pequenas simulações, recorrendo ao simulador de redes *ns-2*, aplicando um simples cenário de configuração de rede e um cenário de tráfego composto pelo envio de apenas uma mensagem de um participante para outro posicionado fora do seu alcance. A rede é composta por cinquenta participantes, cada um com um raio de transmissão de 250 metros e sem mobilidade, colocados aleatoriamente num cenário com as dimensões de $670m \times 670m$. Quando a simulação tem início, o dispositivo emissor deverá enviar uma mensagem a um participante que se encontra fora do seu raio de alcance. Transmite o consequente pedido de rota e todos os dispositivos vizinhos recebem a mensagem e aplicam a política do PAMPA. O resultado está ilustrado na Figura 3.1.

O dispositivo 8 é o emissor e os dispositivos 39, 20, 42, 4 e 46 são os retransmisso-

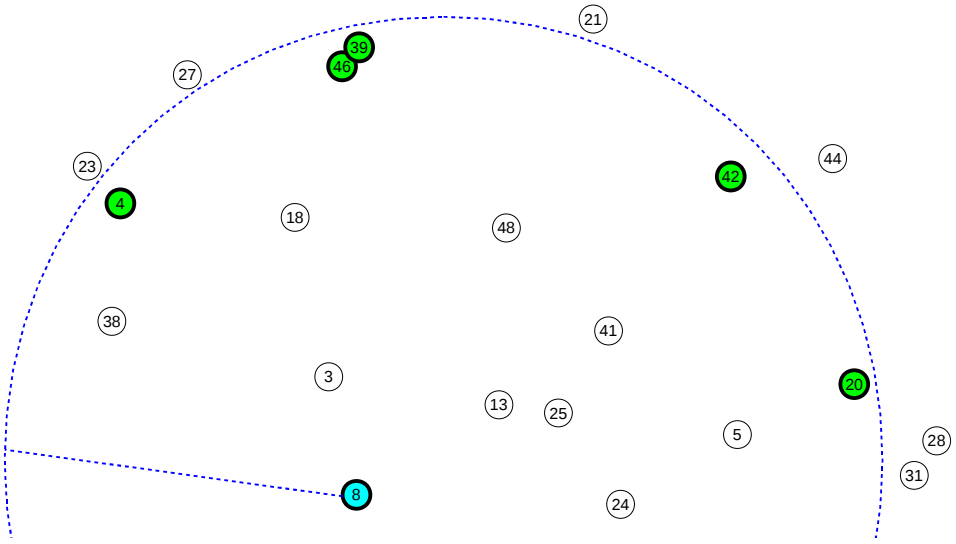


Figura 3.1: Retransmissões do primeiro salto

res [1], enquanto que os restantes participantes dentro do raio de alcance, representado pela circunferência, cancelaram a retransmissão agendada e descartaram a mensagem. Como seria de esperar, os retransmissores são os dispositivos que se encontram mais longe do emissor. Os nós 42, 4 e 46 localizam-se dentro do raio de alcance do nó 39, que retransmite primeiro, pelo que seria, assim, de esperar os respectivos cancelamentos das retransmissões. Porém, a diferença entre as distâncias de cada um, relativas ao emissor, é suficientemente reduzida para que os atrasos calculados sejam demasiado próximos para permitir os cancelamentos esperados.

Ao executar o mesmo teste, utilizando o AODV clássico, todos os dispositivos localizados dentro da circunferência retransmitiram, tal como seria de esperar. O cenário completo é visível na Figura 3.2, onde está assinalada a cobertura proporcionada pelos dispositivos 39, 42 e 4. Os raios de alcance dos nós 20 e 46 não estão assinalados na figura, por não adicionarem cobertura à fornecida pelos outros três. O nó 10 é o participante destinatário, que está assinalado com um quadrado. Como podemos ver, basta a retransmissão do dispositivo 39 para ser descoberta uma rota viável. Note-se que as retransmissões dos dispositivos 39, 42 e 4 são suficientes para disseminar uma mensagem a 76% da rede, isto é, para fazerem chegar uma mensagem a 38 dos 50 dispositivos que compõem esta MANET. Portanto, prevê-se que a aplicação do PAMPA tenha um impacto bastante benéfico no problema de *Broadcast Storm* e no desempenho do AODV, relativamente à aplicação de outros algoritmos de difusão.

[1] As retransmissões destes dispositivos ocorreram pela ordem indicada, ou seja, o nó 39 foi o primeiro a retransmitir

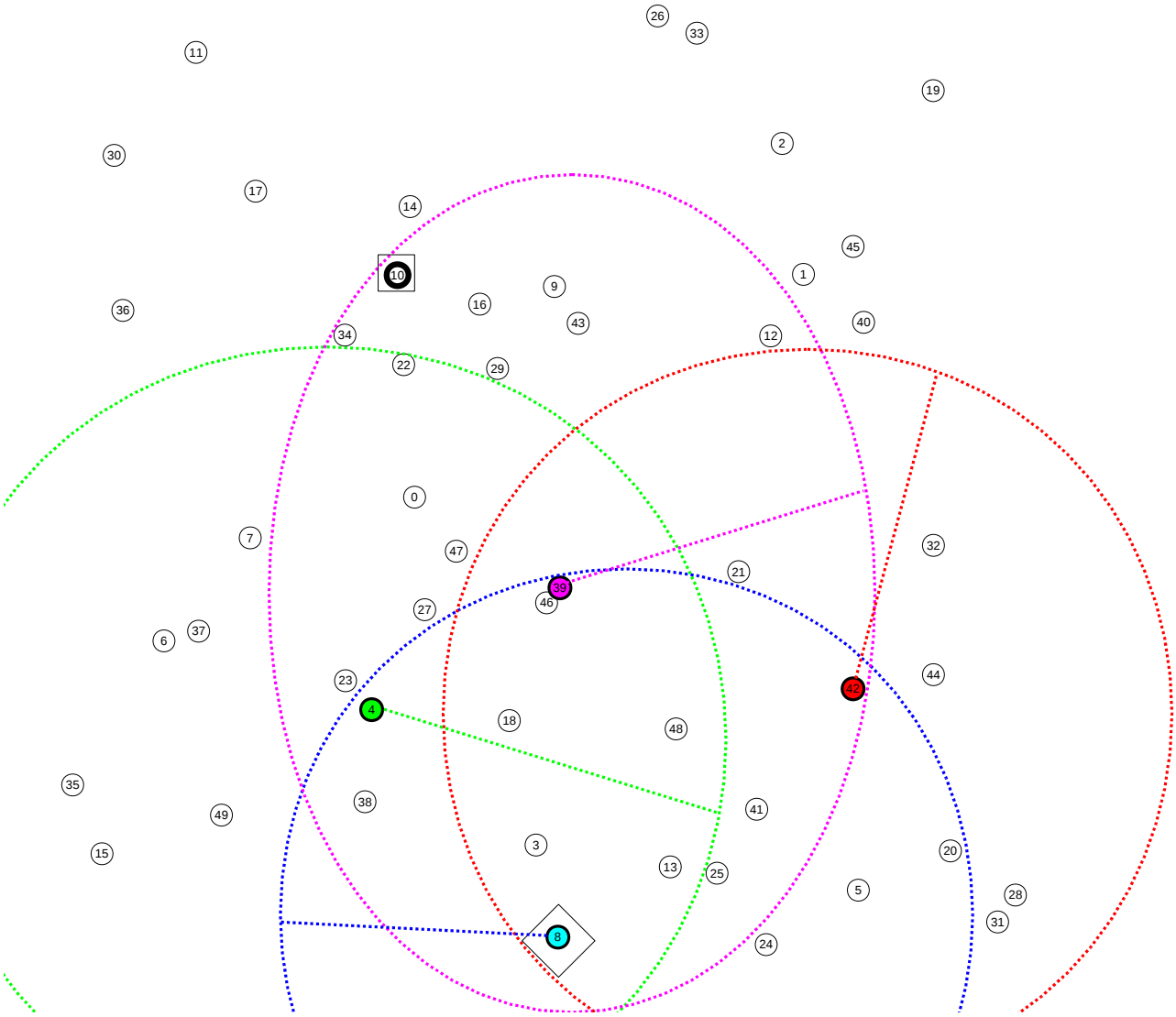


Figura 3.2: AODV+P: Cobertura do primeiro salto

Vulnerabilidade das rotas do PAMPA Considerando ainda a Figura 3.2, uma das rotas possíveis é composta unicamente pelo dispositivo 39 como participante intermédio. Como se pode verificar, este encontra-se no limite do raio de alcance do dispositivo emissor. Tornar as rotas o mais curtas possível, pode ser considerado vantajoso mas, se o participante 39 se deslocar um pouco para norte, a rota torna-se inválida e obriga a que o participante 8 difunda um novo pedido de rota. O problema persiste se a rota escolhida passar pelos dispositivos 20, 42, 4, e 46, já que também se localizam muito perto do limite. Para todos os saltos da disseminação de um pedido de rota, o PAMPA selecciona os dispositivos mais afastados. É por isso que, para cada rota obtida pelo AODV, é grande a possibilidade de incluir pelo menos um dispositivo localizado próximo do limite do raio de alcance do participante que o precede, isto é, as rotas obtidas pelo PAMPA são mais vulneráveis a quebras, quando se verifica muito movimento na rede, do que as obtidas pelo flooding e pelo GOSSIP3(p, k, m).

Tendo em conta que a causa desta vulnerabilidade está relacionada com a selecção dos dispositivos mais afastados do emissor, seleccionar dispositivos mais próximos, soluciona o problema. Por outro lado, quanto mais próximo do emissor for realizada a selecção, menor vai ser a cobertura. Se os dispositivos seleccionados, estiverem localizados ligeiramente antes dos que se encontram no limite do raio de alcance, as rotas seleccionadas serão menos susceptíveis a quebras e os dispositivos que as compõem ainda proporcionam uma cobertura aceitável.

3.3 PAMPA2

Foi desenvolvida uma nova versão do PAMPA, que visa manter as mesmas vantagens e aumentar a tolerância ao movimento dos participantes da rede. Esta versão é de todo semelhante ao PAMPA mas atribui valores de atraso menores aos dispositivos intermédios no raio de alcance do emissor. A característica linear da função do PAMPA, relativa à força de sinal, faz corresponder valores menores de atraso a valores menores de força de sinal, ou seja, ao valor mínimo de força de sinal possível é atribuído o valor mínimo de atraso possível. Sabendo que o valor da força de sinal decai com o aumento da distância, então para o valor máximo possível de distância (dentro do raio de alcance) é atribuído o valor mínimo possível de atraso.

Para esta versão, pretende-se atribuir o valor mínimo possível de atraso a um valor intermédio de intensidade de força de sinal (*IFS*). Este valor intermédio de intensidade de força de sinal deve corresponder a um valor intermédio de distância. Enquanto que a função do PAMPA atraso/IFS é da forma $\mathit{delay}(x) = kx$, onde k é a constante pré-definida, a função alternativa, denominada delay_2, é da forma $\mathit{delay}_2(x) = ax^2 + bx + c$, ou seja, quadrática. Uma função deste tipo é representada graficamente por uma parábola, e assumindo que a sua concavidade é virada para cima, o seu mínimo (isto é, o valor

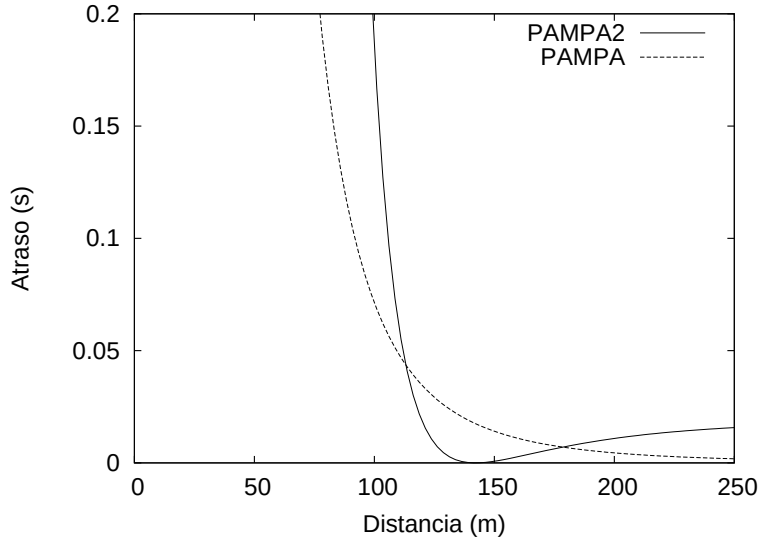


Figura 3.3: Comparação do atraso aplicado pelas funções *delay* e $delay_2$

mínimo da função) corresponde ao seu vértice. Portanto, interessa avaliar e desenvolver uma função atraso/IFS, de segundo grau, cujos valores de a, b e c, colocam o vértice no interior do raio de transmissão. Ou seja, o valor mínimo de atraso corresponde a um valor intermédio de intensidade de força de sinal e, consequentemente, a um valor intermédio de distância. De notar que, por clareza, se utiliza PAMPA e PAMPA2 para distinguir as duas versões da função *delay*.

Foi estudada uma função quadrática para a simulação da placa de rede IEEE 802.11 simulada pelo ns-2 [3], que coloca o seu vértice no interior do raio de alcance dos dispositivos. Pretende-se proporcionar uma tolerância significativa ao movimento dos participantes e uma cobertura aceitável . A função atraso/IFS desenvolvida é dada por $delay_2(x) = \frac{4}{25}(x + 1.5)^2 - \frac{8}{5}(x + 1.5) + 4$. Na Figura 3.3 estão representadas as funções do PAMPA e PAMPA2 em função da distância. Sabendo que o valor da intensidade da força de sinal em função da distância, decai de forma quadrática, no modelo de propagação do *ns-2* então o atraso em função da distância é representado por uma função de segundo grau no PAMPA e por uma função de quarto grau no PAMPA2.

Pode-se observar na Figura 3.3 que o PAMPA2 favorece os participantes que se encontram a uma distância do emissor próxima de 140 metros, enquanto que o PAMPA favorece os participantes que se aproximam dos 250 metros, limite do alcance dos emissores. No entanto, a ideia do PAMPA2 apresenta uma desvantagem. De facto, torna-se necessário saber, antecipadamente, o alcance máximo do raio de transmissão dos dispositivos, isto é, a função $delay_2$ é ideal para dispositivos com um alcance máximo de 250 metros, mas não o é numa rede em que os dispositivos têm um alcance máximo de, por exemplo, 100 metros ou 1000 metros. Assim, as variáveis a, b e c da função do PAMPA2 têm de ser estudadas e determinadas consoante a rede onde se pretende implementar o protocolo.

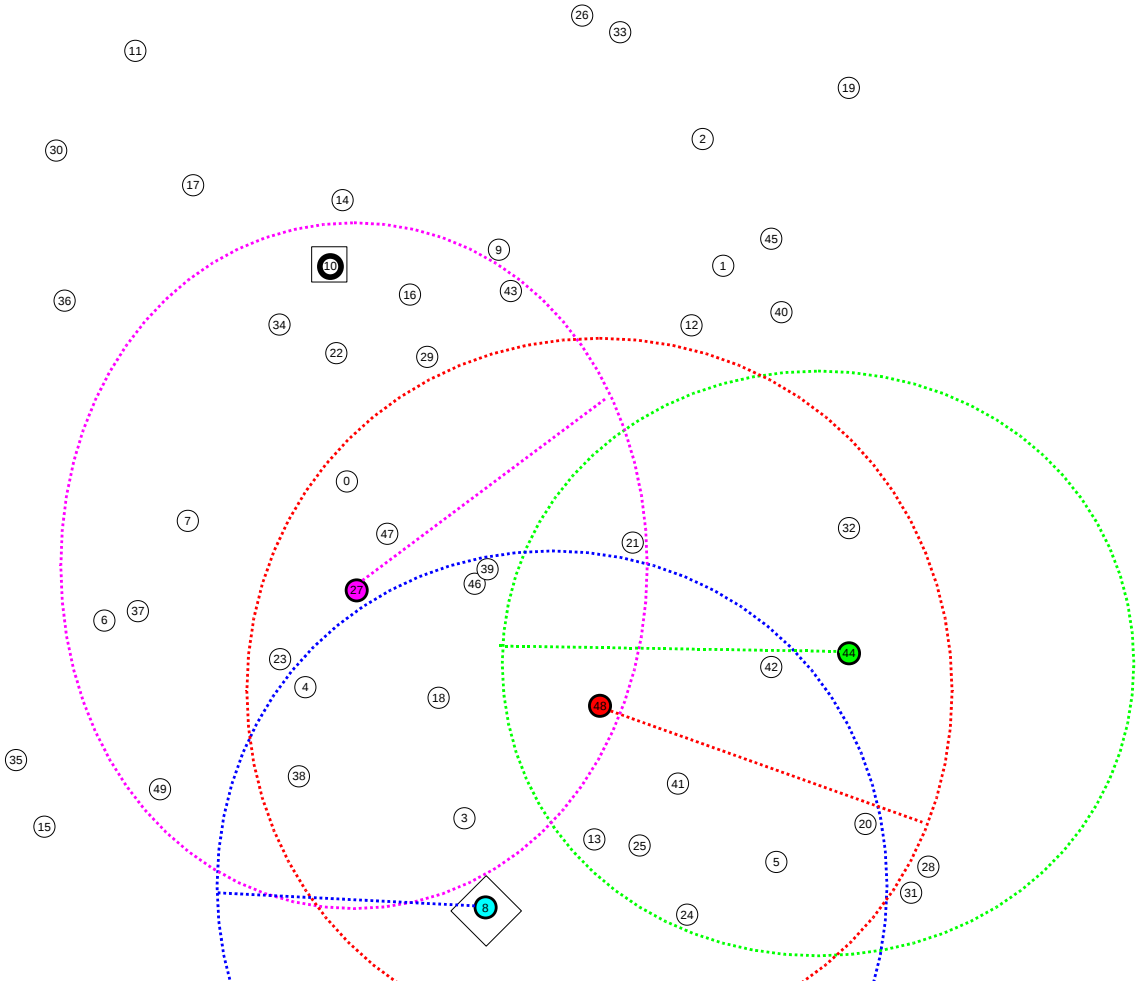


Figura 3.4: AODV+P2 - Cobertura do primeiro e segundo salto

Na Figura 3.4 podemos observar o comportamento do PAMPA2 aplicado ao AODV, sobre as mesmas condições das simulações das anteriores.

Pode-se constatar que, já não são os dispositivos mais afastados dos emissores a retransmitir. A figura permite visualizar que os retransmissores se localizam numa zona menos vulnerável a quebras e que proporcionam uma cobertura bastante razoável. As rotas obtidas neste exemplo nunca impõem menos de dois saltos, ao contrário do AODV+P.

Outro factor a ter em conta, são os atrasos impostos aos participantes mais próximos do emissor. Na Figura 3.3 pode-se constatar que a linha do PAMPA2 é mais inclinada verticalmente do que a linha do PAMPA, o que significa que o PAMPA2 impõe atrasos maiores a participantes localizados num raio de $100m$ do emissor, do que o PAMPA. Ou seja, é possível que a utilização do PAMPA2 seja susceptível a maior latência do que o PAMPA, devido aos casos em que não existem participantes à distância, relativa ao emissor, mais favorecida. A inclinação da linha da função do PAMPA2, para valores de distância à esquerda do vértice, não pode ser optimizada sem alterar as restantes características da função, que proporcionam o comportamento favorável, visualizado na

Figura 3.4. No entanto, poderão ser desenvolvidas funções $delay_2$, com características ligeiramente diferentes da função apresentada, que poderão beneficiar a tolerância ao movimento ou o cálculo dos valores de atraso.

3.4 Sumário

A entrega de pedidos de rota no AODV é concretizada pelo flooding. A utilização do flooding, no AODV, apresenta a vantagem de permitir descobrir todas as rotas existentes entre quaisquer dois dispositivos, proporcionando ao emissor, a possibilidade de escolher a que considerar mais favorável. Por outro lado, o elevado número de retransmissões impostas pelo flooding, pode gerar colisões e, consequentemente, algumas das melhores rotas podem não ser descobertas. É necessário ter em conta que, as perturbações geradas pelo problema de *Broadcast Storm*, causado pelo excesso de retransmissões impostas pelo flooding, pode afectar negativamente o desempenho do AODV, causando, maior latência na descoberta de rotas e na entrega de mensagens, perda de largura de banda e menor taxa de entrega.

A aplicação de algoritmos alternativos de difusão, ao AODV, concedem uma redução do número de dispositivos a retransmitir pedidos de rota, podendo melhorar, portanto, o desempenho do protocolo. No entanto, a redução do número de retransmissões, pode levar a que não sejam encontradas as melhores rotas, já que os dispositivos que as compõem podem não ser seleccionados para retransmitir.

Foi apresentada em [6] a comparação do desempenho do AODV com o flooding e com o GOSSIP3(p, k, m), tendo sido verificado que a instância GOSSIP3$(0.65, 1, 1)$ é benéfica, reduzindo o número de retransmissões de pedidos de rota em 35%, melhorando, consequentemente, o desempenho do AODV, nomeadamente a nível de latência e de taxa de entrega. No entanto, foi observado que as rotas obtidas pelo GOSSIP3$(0.65, 1, 1)$ são mais longas do que as obtidas pelo flooding.

Espera-se que, aplicando o PAMPA ao AODV, o número de retransmissões de pedidos de rota seja menor do que o AODV clássico e que o AODV+G, proporcionando mais benefícios ao desempenho do AODV do que o GOSSIP3(p, k, m). O PAMPA selecciona, dentro do raio de alcance de cada emissor, os participantes mais afastados para retransmitir. Portanto, prevê-se que o AODV+P apresente uma menor tolerância ao movimento dos participantes, derivada do tamanho reduzido das rotas obtidas.

O PAMPA2 propõe remover esta vulnerabilidade das rotas do PAMPA, seleccionando os dispositivos localizados numa distância intermédia dentro do raio de alcance do emissor. Para tal, o PAMPA2 recorre a uma função quadrática, cujo vértice favorece valores de intensidade de força de sinal medianos, para calcular o atraso a ser imposto a uma retransmissão. No entanto, é necessário ter conhecimento, antecipadamente, do raio de alcance dos dispositivos, para implementar esta função.

Capítulo 4

Avaliação

O desempenho do AODV utilizando os algoritmos de flooding, GOSSIP3$(0.65, 1, 1)$, PAMPA e PAMPA2 foi comparado através de simulações. Foi utilizado o simulador de redes $ns-2$ [3]. Pretende-se, assim, avaliar o comportamento do protocolo, quando aplicados os diferentes algoritmos de difusão ao AODV, e perceber qual o impacto de cada um, no tráfego gerado e na qualidade das rotas seleccionadas.

Os cenários de simulação foram baseados nos utilizados em [6]. O modelo de propagação simula uma interface de rede IEEE 802.11, Lucent's WaveLAN [14], com uma largura de banda de $2Mb/s$, alcance de $250m$ e modelo de propagação *two-ray ground* [12].

Todas as simulações têm uma duração de 525 segundos. Em todas, foram dispostos 150 dispositivos, de forma aleatória, numa área de $3300m \times 600m$. Os participantes têm a capacidade de se deslocar livremente numa MANET, de acordo com o modelo de movimento *random waypoint* [7], a uma velocidade uniforme compreendida entre $0m/s$ e $20m/s$. É da maior relevância experimentar diferentes cenários de movimento. Para tal, foram definidos quatro tipos de cenários de mobilidade com tempos de pausa $0s$, $100s$, $300s$ e $500s$.

Tempo de pausa é o intervalo de tempo que cada participante permanece estático, após ter chegado ao primeiro local destino. Ou seja, de acordo com o modelo *random waypoint*[7], cada participante possui uma localização inicial, e uma localização destino, e quando esta é alcançada, o participante pausa durante o tempo definido. Portanto, quanto maior o valor atribuído ao tempo de pausa, menos movimento se verifica na rede. Podemos, por isso, assumir que os dispositivos estão em constante movimento durante os 525 segundos da simulação, nos cenários com tempo de pausa de 0 segundos. Por outro lado, verifica-se muito pouco movimento nos cenários com tempo de pausa de 500 segundos.

Note-se que, atribuindo um valor elevado, como 525 segundos (tempo da simulação), ao tempo de pausa, não implica a inexistência de movimentação na rede. Os participantes possuem uma localização destino, pondo termo à movimentação apenas quando esta for atingida. Ou seja, se um participante definir um local destino, suficientemente longe do local onde se encontra inicialmente, aliado a uma velocidade de deslocação suficiente-

mente reduzida, nunca chega a realizar a pausa. Portanto, mesmo para tempos de pausa muito grandes, existe sempre alguma movimentação na rede.

Em todas as simulações são estabelecidas 30 ligações, a serem mantidas durante todo o tempo de simulação[1]. As ligações são unidireccionais e, os dispositivos que as compõem, são escolhidos aleatoriamente. Foram gerados 160 cenários de tráfego com estas características.

Para garantir a fiabilidade dos resultados, foram gerados 40 cenários de movimento para cada tempo de pausa referido, totalizando 160 cenários de movimento, aos quais foram atribuídos, de forma bijectiva, os 160 cenários de tráfego referidos anteriormente. Recorrendo aos 160 pares <cenário de tráfego, cenário de movimento> foram realizadas 160 simulações para cada algoritmo de difusão (referidos anteriormente) aplicado ao AODV.

Para o expanding-ring search do AODV, são utilizados os valores de origem. É atribuído um tempo de vida inicial de cinco saltos às mensagens de pedido de rota. Se não for obtida resposta, repete-se o processo, atribuindo um tempo de vida de sete saltos. Caso não seja encontrada uma rota, a mensagem é difundida pela rede toda. A difusão das mensagens através do expanding-ring search é assegurada pelo algoritmo de difusão em avaliação.

A definição dos valores dos parâmetros do PAMPA e PAMPA2, constante e *treshhold*, é da maior relevância. Relativamente à constante, um valor atribuído, demasiado pequeno, pode levar a que ocorram mais retransmissões do que se pretende, porque os valores de atraso calculados pelos retransmissores, são demasiado próximos. Por outro lado, valores muito grandes atribuídos à constante, levam ao calculo de valores de atraso demasiado elevados. Foi atribuído o valor $5\text{x}10^6$, por ter sido considerado ideal em [9]. Foi atribuído o valor 1 à variável de *treshold* do PAMPA e PAMPA2, isto é, os participantes descartam uma mensagem em fila de espera após receber uma cópia da mesma.

A equidade entre algoritmos é garantida pela aplicação dos mesmos pares <cenário de tráfego, cenário de movimento> a todos os algoritmos. Os valores apresentados são a média aritmética dos resultados das 40 simulações de cada algoritmo em cada cenário.

4.1 Discussão sobre os parâmetros de simulação

Os parâmetros de simulação escolhidos, são iguais aos utilizados em [6]. Em preparação para as simulações, vale a pena perceber de que forma estes parâmetros influenciam os resultados.

[1] Ou seja, se a rota, que se encontra a ser utilizada pela ligação, for quebrada, o emissor inicia uma operação de descoberta de rota para o mesmo participante destinatário

4.1.1 Modelo de movimento *random waypoint*

O modelo de movimento *random waypoint* foi criado com o objectivo de simular movimento aleatório de participantes numa MANET [7]. São escolhidas, aleatoriamente, as coordenadas da localização destino para cada participante. Os participantes deslocam-se para as respectivas localizações destino a uma velocidade escolhida, também aleatoriamente, entre um valor mínimo e um valor máximo. Quando um participante chega ao local destino, permanece estático durante o tempo de pausa, definido para a simulação, e repete o processo. Isto é, determina novas coordenadas de localização destino e uma nova velocidade de deslocamento. De notar que, cada participante escolhe aleatoriamente a sua velocidade de movimento.

Os locais de destino e a velocidade de movimento, são de facto, determinados aleatoriamente, para cada participante. No entanto, foi demonstrado que a utilização deste modelo pode apresentar alguns problemas nestes aspectos. A média aritmética das velocidades dos dispositivos de uma simulação é, tipicamente, muito próxima de metade do valor de velocidade máxima e, com o decorrer do tempo de simulação, tende a decrescer [15]. Para tempos de simulação grandes, o valor desta média chega a atingir valores muito próximos de $0m/s$. Se um participante atribuir um valor elevado à velocidade de deslocamento, atinge a localização destino em pouco tempo de simulação e, se atribuir um valor reduzido à velocidade de deslocamento, necessita de mais tempo para atingir a localização destino.

A probabilidade de um participante escolher uma velocidade próxima do valor máximo é igual à probabilidade de escolher uma velocidade próxima do valor mínimo, mas para valores maiores escolhidos, a localização destino é alcançada mais rapidamente. Consequentemente, o participante rapidamente volta a ter de escolher um novo valor de velocidade de deslocamento. Ou seja, a probabilidade de ser escolhido um valor baixo de velocidade, durante a simulação, é elevada e, tal como foi referido anteriormente, vai ser atribuído mais tempo de simulação para esta deslocação, comprometendo o valor da média. Este factor justifica, também, porque o valor da média decai com o decorrer do tempo de simulação.

Outro problema deste modelo, é a tendência de haver uma maior densidade de dispositivos no centro da área, com o decorrer do tempo de simulação [4]. Para todas as localizações iniciais possíveis, a probabilidade, da trajectória interceptar a zona central da área de simulação, é muito elevada. Portanto, observando a disposição dos participantes na área de simulação, durante a evolução da simulação, pode-se visualizar um aglomerado de participantes no centro da área. Este fenómeno representa um problema sério para alguns algoritmos de difusão, como por exemplo, o flooding e o GOSSIP3(p, k, m). O número elevado de retransmissões, impostas pelo flooding, aliado à elevada concentração de dispositivos no centro, origina um agravamento considerável do problema de *Broadcast Storm*.

No caso do GOSSIP3(p, k, m), verifica-se o mesmo problema, devido à sua incapacidade de adaptação a diferentes condições numa rede. Ou seja, assumindo que a rede começa com uma distribuição homogénea dos participantes, não se espera a atribuição de um valor baixo ao parâmetro p e sabendo que a distribuição dos participantes na rede tem tendência a tornar-se heterogénea, prevê-se o mesmo problema. Por outro lado, o PAMPA e o PAMPA2, são muito pouco afectados por este problema, pois o seu comportamento é sempre o mesmo, independentemente da distribuição dos dispositivos[9].

4.1.2 Tráfego imposto na rede

É da maior relevância analisar o tráfego imposto na rede, consequente das características das simulações realizadas. Em todas as simulações, são enviados dois pacotes por segundo por cada uma das 30 ligações, o que resulta em 60 pacotes por segundo, no total. Vai ser observado, na secção 4.3, que o valor típico do comprimento médio das rotas obtidas pelo AODV (excepto o AODV+P, que apresenta rotas consideravelmente mais curtas) é de 6 saltos. Assumindo este valor como referência para número de saltos e sabendo que são enviados 60 pacotes por segundo no total, então ocorrem 360 transmissões por segundo, para a rede toda. Ou seja, são enviados 30720 bytes por segundo, uma vez que o tamanho de cada pacote é 512 bytes.

Para todas as simulações, a largura de banda é de $2Mb/s$, mas se duas transmissões concorrentes ocorrem em simultâneo, enquanto os dois participantes em causa estão afastados por mais de $250m$, então ambos usufruem da largura máxima ($2Mb/s$). Portanto, sabendo que os dispositivos possuem uma área de alcance de $196250m^2$ (isto é, para um raio de alcance de $250m$, a área é $\pi \text{x} 250^2$), numa área de simulação com $1980000m^2$ (isto é, $3300m \times 600m$), é possível (no melhor caso possível, ou seja, com uma distribuição dos 150 dispositivos da forma mais favorável) que 10 participantes transmitam em simultâneo e usufruam da $2Mb/s$ de largura de banda. Sabendo que ocorrem 360 transmissões por segundo, então, para cada uma das 10 regiões referidas, ocorrem 36 transmissões por segundo.

Mesmo nas condições muito favoráveis apresentadas, o tráfego representa já uma carga não negligenciável na rede. Os problemas de congestão na rede são amplificados por muitas destas retransmissões terem que partilhar a mesma região, gerando por isso situações de competição pelo acesso ao meio por diferentes dispositivos. Este efeito deverá ser sobretudo evidente no centro do espaço da simulação, região que, com maior probabilidade, será atravessada por um maior número de rotas.

4.2 Distribuição de tráfego por participantes

No AODV, os participantes emissores não têm em conta a equidade da distribuição do tráfego pelos participantes da MANET, durante selecção das rotas a serem utilizadas para

transferências de dados. Consequentemente, prevê-se que alguns dos dispositivos tenham uma carga adicional de operações de retransmissão, comparativamente a outros dispositivos.

O número total de mensagens retransmitidas na rede, tem um impacto relevante no desempenho do AODV. Adicionalmente, a distribuição das operações de retransmissão por todos os participantes da rede, também é da maior importância neste âmbito. Este factor reflecte-se numa distribuição desigual do consumo de recursos energéticos, comunicacionais e computacionais, pelos participantes da MANET. Por exemplo, é possível que, em determinados instantes, alguns dispositivos tenham pouca bateria disponível, enquanto que outros têm a sua carregada quase totalmente.

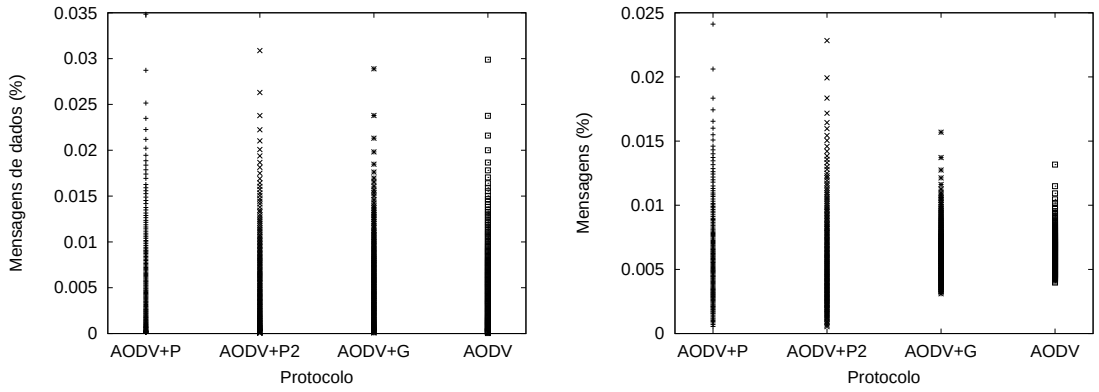
Outra consequência que deve ser considerada é o impacto de uma distribuição de tráfego desigual, no desempenho do AODV, a nível de estabilidade de rotas, latência e taxa de entrega. Se um pequeno número de dispositivos são incluídos num grande número de rotas em simultâneo, espera-se alguma congestão, localmente, nesses dispositivos. Uma rota pode ser dada como inválida, devido a um atraso, consequente de uma congestão presente num dos dispositivos que compõem a rota.

Portanto, uma distribuição desigual de tráfego pelos participantes, pode levar a operações de descoberta de rota adicionais, bem como, aumento de latência e perda de mensagens de dados.

Para cada uma das 160 simulações, realizadas para cada algoritmo, foi contabilizado o número de retransmissões de pedidos de rota e de mensagens de dados, em cada um dos 150 dispositivos.

Os resultados, de cada uma das quatro versões do AODV estudadas neste trabalho, foram agrupados pelos quatro tempos de pausa. Em cada uma das 40 simulações (de cada tempo de pausa) foi seleccionado o dispositivo mais sobrecarregado e, para o conjunto de dispositivos resultante, foi feita a média aritmética do número de mensagens de dados, bem como a média aritmética da soma do número de pedidos de rota com mensagens de dados. O mesmo processo foi repetido para os restantes 149 dispositivos de cada uma das 40 simulações (segundo dispositivo mais sobrecarregado de cada simulação, terceiro, até ao menos sobrecarregado).

Para cada versão do protocolo, AODV, AODV+G, AODV+P, AODV+P2, é apresentada uma distribuição média de tráfego (em percentagem), de mensagens de dados e de mensagens de pedidos de rota juntamente com mensagens de dados, para cada tempo de pausa. Os gráficos que contêm a distribuição de transmissões de mensagens de dados, permitem confirmar, se alguns dos dispositivos da rede são seleccionados para compor rotas, com mais frequência do que outros. As figuras que apresentam a distribuição de transmissões de pedidos de rota juntamente com transmissões de dados, permitem verificar até que ponto, alguns dispositivos são mais sobrecarregados com transmissões do que os restantes.



(a) Mensagens de dados

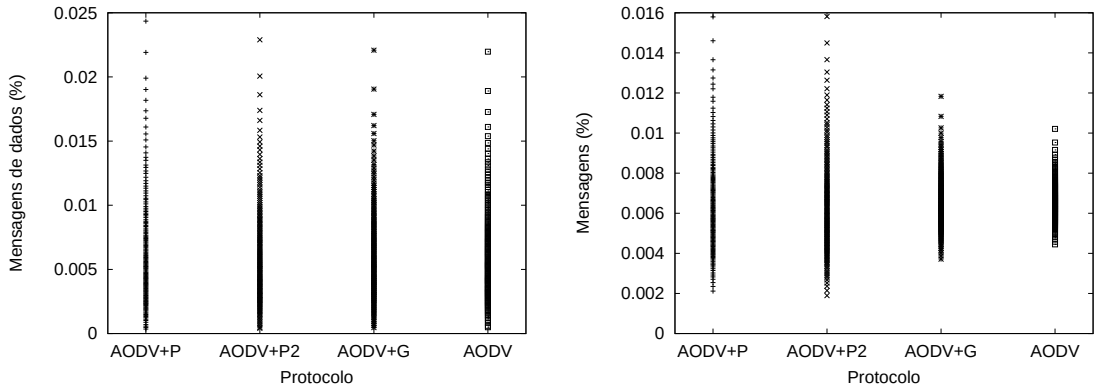
(b) Mensagens de dados e pedidos de rota

Figura 4.1: Distribuição de tráfego para 500 segundos de tempo de pausa

Pouca movimentação na rede As duas figuras apresentadas em 4.1 mostram a distribuição de tráfego média dos 40 cenários de 500 segundos de tempo de pausa, para cada uma das quatro versões do protocolo. A Figura 4.1(a) ilustra a distribuição de mensagens de dados, enquanto que a Figura 4.1(b), ilustra a distribuição de mensagens de dados em conjunto com mensagens de pedidos de rota.

Pode-se observar na Figura 4.1(a) que, em todas as versões do AODV, alguns dispositivos enviam bastante mais mensagens de dados do que os restantes participantes. Se tivermos em conta que, 0.035% equivale a, aproximadamente, 3600 mensagens, então, para todas as versões do protocolo, verifica-se a existência de, aproximadamente, 5 a 10 dispositivos que retransmitem mais de 2000 mensagens e um aglomerado de participantes que retransmitem menos de 1000 mensagens. Verifica-se também, em todas as versões, a existência de participantes que não retransmitem mensagens de dados, ou retransmitem muito poucas. Ou seja, as zonas de maior densidade de dispositivos, nesta figura, estão compreendidas entre os valores 0% e 0.01% (0 e 1000 mensagens), observando-se alguns dispositivos, não mais do que uma dezena, compreendidos entre 0.02% e 0.035% (2000 e 3600 mensagens).

A distribuição desigual verificada na Figura 4.1(a), mantém-se evidente na Figura 4.1(b). Porém, na versão clássica do AODV, todos os dispositivos retransmitem mais de 0.015% do número total de mensagens (aproximadamente), pois o flooding obriga a que todos os participantes retransmitam as mensagens recebidas pela primeira vez. Ou seja, se o gráfico mostrasse apenas as mensagens de pedidos de rota, observar-se-ia uma distribuição com equidade por parte do AODV clássico, mas significaria apenas que todos os participantes são sobrecarregados em vez de alguns. O AODV+G apresenta dispositivos a retransmitirem menos mensagens do que os dispositivos menos sobrecarregados do AODV, pois o GOSSIP3$(0.65, 1, 1)$ proporciona uma redução considerável de retransmissões de pedidos de rota relativamente ao flooding. No entanto, todos os participantes retransmitem mais do que 0.003% (1000 mensagens, aproximadamente) com o AODV+G, o que



(a) Mensagens de dados (b) Mensagens de dados e pedidos de rota

Figura 4.2: Distribuição de tráfego para 300 segundos de tempo de pausa

não se verifica com o AODV+P e AODV+P2. Estas 3 versões proporcionam reduções de retransmissões de pedidos de rota, relativamente ao AODV, mas o AODV+G concretiza as suas reduções, de forma probabilista, sendo extremamente improvável que um dispositivo da rede nunca seja seleccionado para retransmitir uma mensagem. Por outro lado, o AODV+P e o AODV+P2 concretizam as reduções com base em distância, sendo, por isso, possível que alguns dos dispositivos nunca retransmitam uma mensagem, ou retransmitam muito poucas mensagens, o que pode ser constatado na Figura 4.1(b).

Nas duas figuras apresentadas em 4.2, pode-se observar que a distribuição do tráfego é ligeiramente menos desigual do que nas figuras em 4.1. Isto é, tendo em conta a escala de cada um dos gráficos, nota-se um pouco menos de dispersão dos respectivos pontos. Os dispositivos mais sobrecarregados em 4.1, encontram-se mais afastados da zona mais concentrada, do que em 4.2. Isto sugere menor desigualdade na distribuição de tráfego em 4.2 do que em 4.1.

Nestes cenários, os participantes mantêm-se estáticos durante menos tempo, comparativamente aos cenários de tempo de pausa de 500 segundos, logo, os dispositivos mais sobrecarregados mantêm-se menos tempo nas localizações geográficas que os levam a ser seleccionados para retransmitir com mais frequência.

O aumento do movimento, torna improvável a existência de dispositivos que não retransmitam (ou retransmitam pouco) no AODV+P e AODV+P2, o que não se verifica nos cenários de 500 segundos de tempo de pausa. Podemos observar na Figura 4.2(b) que os dispositivos menos sobrecarregados, enviam mais mensagens do que na Figura 4.1(b), confirmando, por isso, a afirmação anterior.

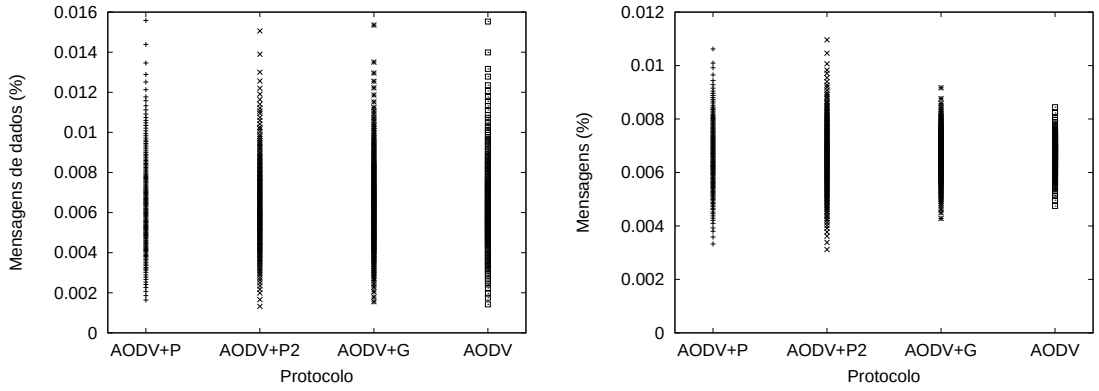
Muita movimentação na rede Com o aumento do movimento, espera-se uma distribuição de tráfego menos desigual, comparativamente aos cenários que apresentam pouco movimento. Ou seja, os participantes encontram-se em movimento, durante a maioria do tempo de simulação, impondo por isso, alguma aleatoriedade na distribuição de tráfego.

Pode ser observado nas figuras apresentadas em 4.3 que todos os dispositivos enviam mensagens, tanto mensagens de dados (Figura 4.3(a)) como mensagens de pedidos de rota (Figura 4.3(b)), o que nem sempre era verificado nos cenários de pouco movimento.

Nestas figuras nota-se, em todas as versões do AODV, uma concentração maior de dispositivos, relativamente aos cenários de pouco movimento, apresentados nas figuras 4.2 e 4.1. Por exemplo, para o AODV+P2, na Figura 4.2(a) observámos uma dispersão entre 0% mensagens e 0.025% (0 e 2500 mensagens, aproximadamente) e na Figura 4.3(a) observamos uma dispersão entre 0.002% e 0.016% (100 mensagens e 1400 mensagens, aproximadamente). Isto sugere que, os cenários de pouco movimento são mais susceptíveis a uma maior desigualdade na distribuição de tráfego. Este facto pode ser observado, com mais clareza, nas imagens de distribuição de tráfego, para as simulações sem tempo de pausa, apresentadas nas figuras 4.4(a) e 4.4(b). Nestas figuras é evidenciada uma dispersão muito menor das mensagens retransmitidas pelos dispositivos do que nas figuras visualizadas anteriormente. Esta maior concentração de dispositivos, confirma as suspeitas levantadas, relativamente ao impacto da movimentação dos participantes, na distribuição de tráfego.

Porém, mesmo com a movimentação dos participantes durante a totalidade do tempo de simulação, o problema da desigualdade de distribuição do tráfego, mantém-se, como pode ser constatado nas figuras ilustradas em 4.4.

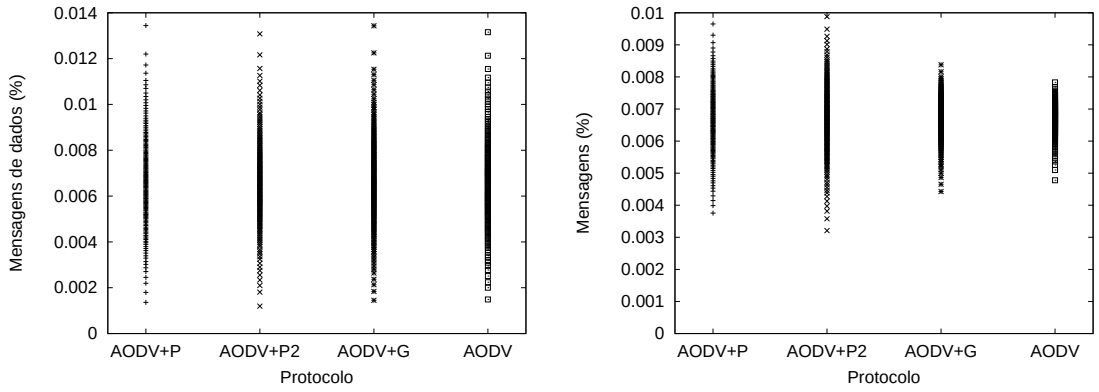
O critério de selecção de rotas do AODV proporciona a escolha das mais favoráveis para o emissor, sem qualquer ponderação do impacto na distribuição de tráfego na rede. O tráfego imposto na rede, discutido na sub-secção 4.1.2, também tem um grande impacto nos resultados obtidos. Relativamente à difusão de pedidos de rota, o PAMPA, PAMPA2 e GOSSIP3(p, k, m) também não ponderam este aspecto. O flooding obriga a que todos retransmitam sempre os pedidos de rota recebidos pela primeira vez, o que não deve ser visto como uma distribuição com equidade, mas sim como uma sobrecarga



(a) Mensagens de dados

(b) Mensagens de dados e pedidos de rota

Figura 4.3: Distribuição de tráfego para 100 segundos de tempo de pausa



(a) Mensagens de dados

(b) Mensagens de dados e pedidos de rota

Figura 4.4: Distribuição de tráfego para 0 segundos de tempo de pausa

adicional imposta nos dispositivos, cujas retransmissões implicam redundância. Portanto, este problema de distribuição, está presente nas quatro versões do AODV apresentadas, podendo, aliado a um tráfego excessivo que poderá ser imposto na rede, reflectir-se numa degradação do desempenho do protocolo.

4.3 Estabilidade das Rotas

O número de pedidos de rota solicitados pelos dispositivos em cada um dos algoritmos é apresentado na Figura 4.5. Podemos observar que o número de operações de descoberta de rota diminui com o aumento do tempo de pausa, para todos os algoritmos. Seguramente, podemos assumir que, quanto maior é o tempo de pausa, menos movimento se verifica na rede por parte dos participantes. Assumindo que o movimento dos participantes é a principal causa de quebra de rotas no AODV, então facilmente se justifica esta diminuição.

Nesta figura observa-se um número demasiado elevado de operações de pedidos de rota, quando se tem em conta apenas 30 ligações durante 525 segundos de tempo de simulação. De facto, a movimentação dos participantes na rede, pode ser considerado insuficiente para justificar números de pedidos de rota superiores a 4500. Note-se que, para 500 segundos de tempo de pausa, isto é, cenários com pouco movimento, são realizados entre 2500 e 3500 pedidos de rota. Considerando 3000 pedidos de rota como referência, são realizados, em média, 100 por cada uma das 30 ligações. Sabendo que o tempo de simulação é de 525 segundos, então, em média, é enviado um pedido de rota em cada 5 segundos.

As suspeitas levantadas anteriormente, relativas às condições agrestes de tráfego imposto na rede, bem como ao problema de distribuição desigual de tráfego (discutido na secção 4.2), podem justificar os resultados ilustrados na Figura 4.5. Ou seja, a congestão,

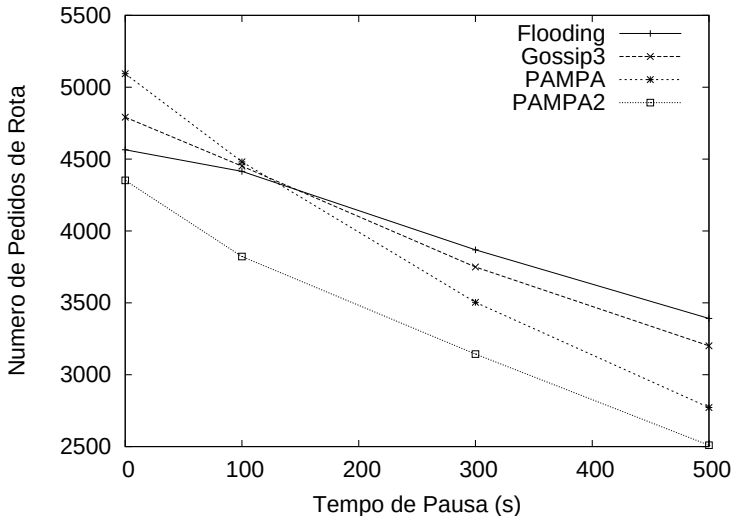


Figura 4.5: Operações de descoberta de rota

que poderá ser causada pelo conjunto destes dois factores, leva a que muitas rotas sejam dadas como inválidas e, consequentemente, são difundidos mais pedidos de rota do que se esperaria.

A figura também confirma as dúvidas levantadas no capítulo anterior sobre a qualidade das rotas seleccionadas, em particular, a baixa tolerância do PAMPA ao movimento dos participantes. Na verdade, com PAMPA, os participantes realizam mais pedidos de rota com tempos de pausa iguais ou inferiores a 100 segundos, comparativamente aos outros algoritmos. Por outro lado, para tempos de pausa iguais ou superiores a 300 segundos, os participantes difundem menos pedidos de rota com o PAMPA do que com o flooding e com o GOSSIP3(p, k, m). Isto porque a ocorrência de quebra de rotas é aliviada com a diminuição do movimento na rede e o PAMPA impõe uma redução de retransmissões de pedidos de rota, comparativamente ao Flooding e ao GOSSIP3(p, k, m) (o que vai ser conferido na secção 4.4). Esta redução atenua a congestão, consequente dos problemas de distribuição de tráfego e excesso de tráfego, discutidos anteriormente.

Com o PAMPA2, o AODV realiza sempre menos pedidos de rota do que com qualquer um dos outros, visto que, por um lado, tal como o PAMPA, proporciona reduções a nível de retransmissões de pedidos de rota, comparativamente ao flooding e ao GOSSIP3(p, k, m) e, por outro lado, não é tão vulnerável a quebra de rotas como o PAMPA.

A vulnerabilidade, definida pelo comprimento reduzido das rotas do PAMPA, pode ser visualizada na Figura 4.6, que apresenta o número médio de saltos, podendo-se constatar que a linha do PAMPA está completamente separada das dos restantes algoritmos. O PAMPA apresenta valores entre 4.6 e 5.9 (aproximadamente), enquanto que os outros 3 algoritmos estão compreendidos entre 5 e 6.4 (aproximadamente).

O aumento do comprimento das rotas consoante o aumento do tempo de pausa, confirma a tendência do modelo de movimento de *random waypoint* de concentrar os dis-

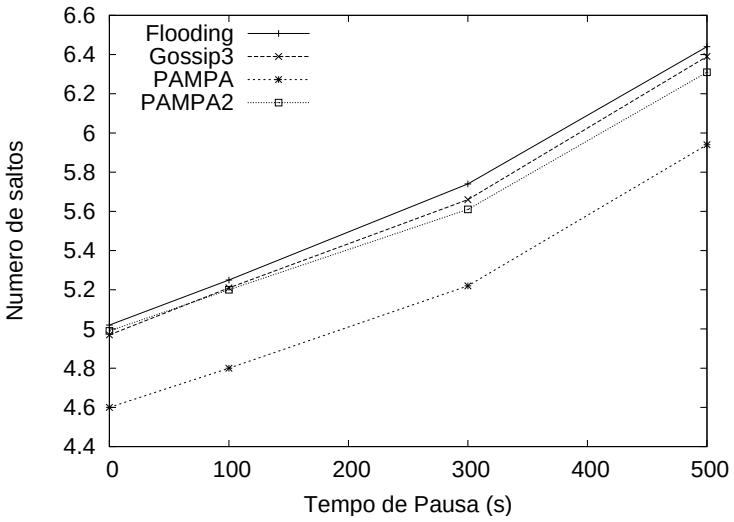


Figura 4.6: Comprimento médio das rotas

positivos no centro da área simulada. Ou seja, para os cenários que apresentam mais movimento, a concentração de participantes no centro da área de simulação é maior (logo a proximidade entre os participantes é maior) e, consequentemente, as rotas obtidas pelos participantes são mais curtas.

4.4 Tráfego de Pedidos de Rota

A Figura 4.7 apresenta o acumulado de retransmissões de route requests, realizadas por todos os dispositivos. Como se pode observar, o PAMPA e o PAMPA2 reduzem, de uma forma massiva, o número de retransmissões na rede relativamente ao flooding. O PAMPA concretiza reduções de tráfego até 72% e o PAMPA2 até 74% em relação ao flooding, facto que se pode constatar na Figura 4.8. Em comparação com o GOSSIP3(p, k, m) os ganhos atingem 55% para o PAMPA e 59% para o PAMPA2.

É necessário ter em conta que as métricas avaliadas na secção anterior, influenciam directamente os valores de tráfego total, isto é, comprimento médio das rotas e número de operações de descoberta de rota. É intuitivo assumir que uma rota mais longa, implica mais retransmissões. O mesmo se pode afirmar do número de pedidos de rota efectuados, ou seja, mais pedidos de rota origina mais retransmissões.

Observando a Figura 4.7, podemos constatar que o número de retransmissões nos cenários diminui com o aumento do tempo de pausa, para todos os algoritmos de difusão. A diminuição da mobilidade dos participantes na rede influenciou também este aspecto, ao serem gerados menos pedidos de rota, visto que diminuiu a necessidade de as renovar.

O AODV+P realiza sempre mais operações de descoberta de rota do que o AODV+P2, por ser mais susceptível a quebras de rota. Este facto constatado anteriormente explica a ligeira diferença entre as reduções impostas pelo PAMPA e pelo PAMPA2, podendo

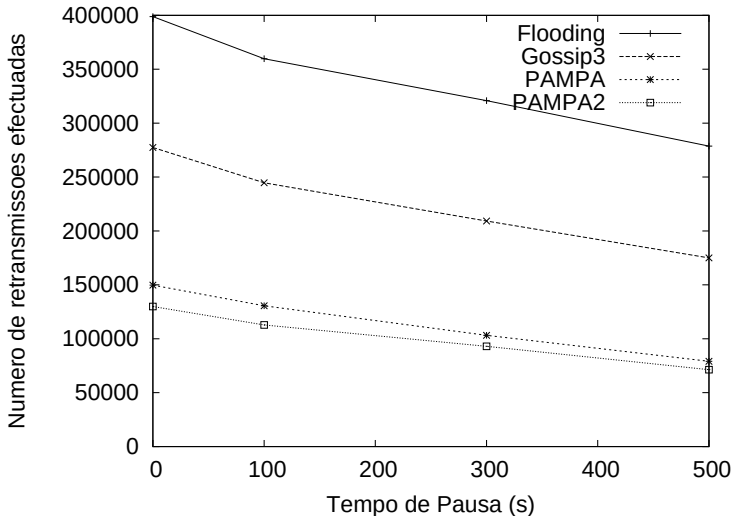


Figura 4.7: Total de retransmissões de pedidos de rota

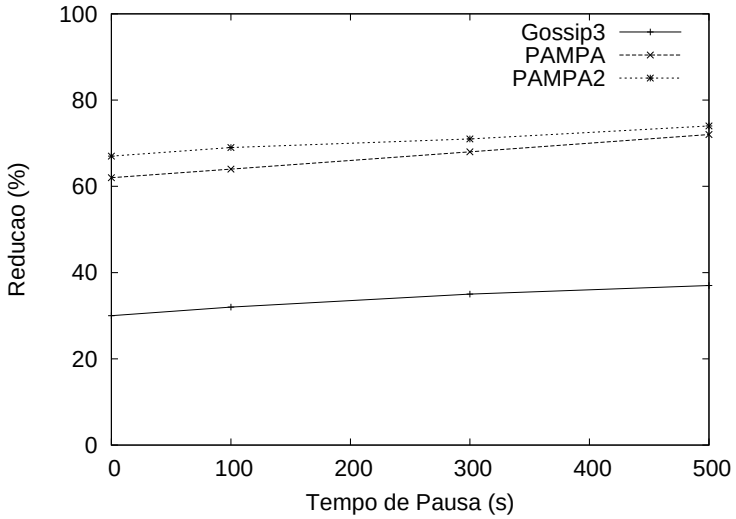


Figura 4.8: Redução de retransmissões de pedidos de rota relativamente ao flooding

observar-se que é maior no PAMPA2, ou seja, mais pedidos de rota implicam mais retransmissões a ter em conta. Apesar do AODV+P2 apresentar um comprimento médio de rota maior do que o AODV+P (Figura 4.6), esta diferença tem um impacto mais pequeno, na redução do número de retransmissões, comparativamente à diferença de número de pedidos de rota entre os dois algoritmos.

4.5 Latência

A significativa redução de tráfego observada tem um impacto considerável na latência. Os tempos de descoberta de rota e os tempos de entrega de mensagens de dados, podem ser observados nas figuras 4.9 e 4.10.

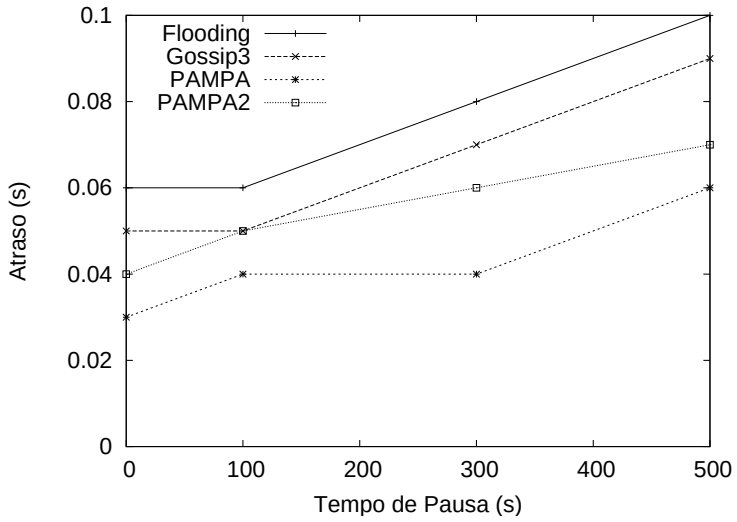


Figura 4.9: Latência nas operações de descoberta de rota

O tempo de descoberta de rota foi medido pelo intervalo de tempo compreendido entre, o instante em que um pedido de rota é transmitido pelo emissor e o instante em que a primeira resposta (respectiva a este pedido de rota) lhe é entregue. Relativamente à medição do tempo de envio de mensagens de dados, é necessário ter em conta que, se uma rota para o destinatário não estiver disponível no momento do envio, o intervalo de tempo consumido na descoberta da respectiva rota, é incluído na medição.

A ocorrência de colisões, durante uma operação de descoberta de rota, pode levar a que o respectivo pedido de rota, tenha de ser repetido. Tendo em conta, a forma como o intervalo de tempo de descoberta de rota é medido, a ocorrência de colisões pode afectar nitidamente os tempos obtidos.

Intuitivamente se assume que problemas de congestão e de contenção nas retransmissões, influenciam, directamente, os valores de latência obtidos.

Observando as figuras, podemos verificar que o AODV+P e o AODV+P2 conseguem melhores tempos, comparativamente ao AODV e ao AODV+G, tanto a nível de operações de descoberta de rota, como a nível de entrega de dados. A redução da ocorrência de colisões, contenções de transmissão e congestão, proporcionada pela enorme diminuição de dispositivos a retransmitir, apresentada nas figuras 4.7 e 4.8, justifica os ganhos referidos.

De notar que o PAMPA2 impõe maior latência ao AODV do que o PAMPA. Isto porque a função $delay_2$, desenvolvida para o PAMPA2, impõe valores de atraso maiores do que se desejaria, aos participantes que não se encontram na distância mais favorecida pela função. A Figura 3.3, apresentada no Capítulo 3, ilustra esta comparação entre as funções do PAMPA e PAMPA2. Ainda assim, os tempos oferecidos pelo PAMPA2 são mais favoráveis do que os proporcionados pelo flooding e GOSSIP3(p, k, m).

Com o aumento do tempo de pausa e a consequente diminuição do movimento dos participantes, é reduzida a ocorrência de quebra de rotas, para todos os algoritmos. Este

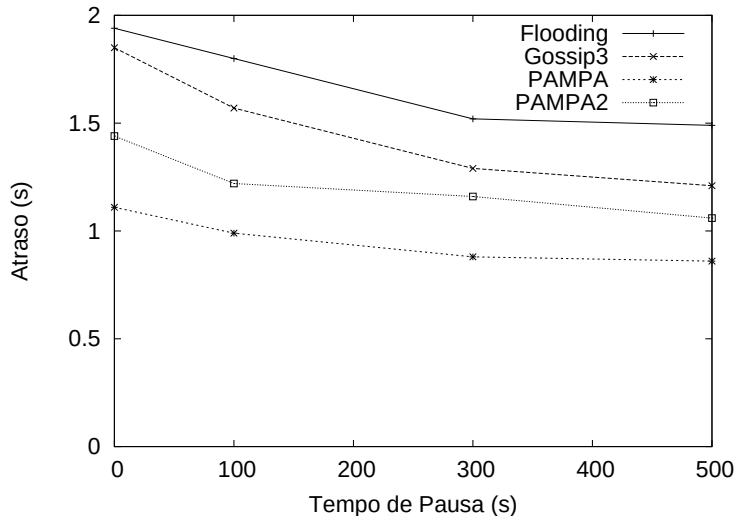


Figura 4.10: Latência na entrega de dados

facto, já confirmado na Figura 4.5, é a razão pela qual a latência na entrega de mensagens de dados, apresentada por todos os algoritmos, diminui com o aumento do tempo de pausa.

Intuitivamente, uma rota mais comprida leva mais tempo a ser obtida, por implicar um maior número de saltos na propagação do pedido de rota e também no envio da resposta. Portanto, a subida do comprimento médio das rotas, consoante o aumento do tempo de pausa visualizado anteriormente na Figura 4.6, contribui para o crescimento da latência, verificado nas operações de descoberta de rota, com o aumento do tempo de pausa. Adicionalmente, a existência de congestão na rede, consequente dos problemas de excesso e distribuição de tráfego, sugeridos anteriormente, contribui fortemente para justificar este aumento de latência relativo ao aumento do tempo de pausa.

4.6 Taxa de entrega

Esperava-se um impacto significativo na taxa de entrega do AODV, quando sujeito à aplicação do PAMPA e do PAMPA2, devido à enorme redução de retransmissões imposta na rede. Este impacto benéfico pode ser visualizado na Figura 4.11. O AODV+P e o AODV+P2, têm uma taxa de entrega compreendida entre 60% e 72%, aproximadamente. O AODV+G possui uma taxa de entrega compreendida entre 51% e 70% e o AODV apresenta 45% e 65%.

A ocorrência de quebras de rota, é uma causa de perda de mensagens. O emissor envia 2 mensagens por segundo e, se a rota for quebrada entretanto, as mensagens enviadas, até o participante emissor receber uma mensagem de *route error*, são perdidas.

A vantagem do PAMPA e do PAMPA2, sobressai nos valores mais reduzidos de tempo de pausa e é atribuída à redução do tráfego proporcionada pelos dois algoritmos. Ou seja,

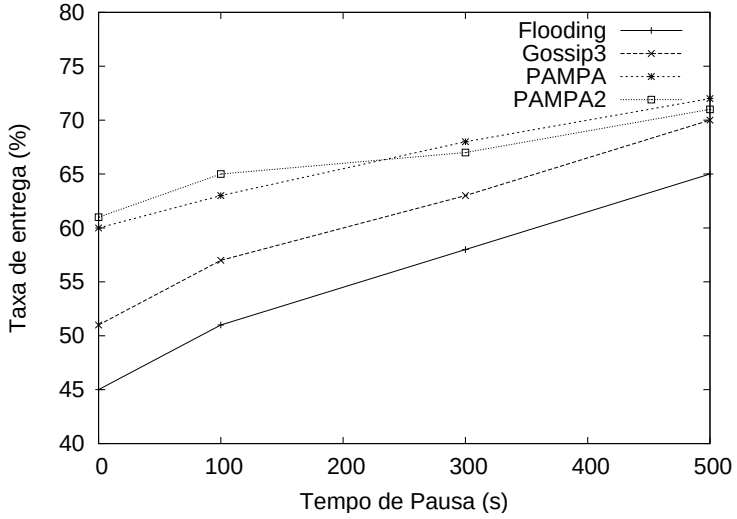


Figura 4.11: Taxa de entrega

para tempos de pausa menores, verificam-se mais operações de descoberta de rota, e consequentemente mais colisões e congestão, destacando-se, por isso, as vantagens do PAMPA e do PAMPA2. Apesar do PAMPA ser o mais susceptível a quebras de rota, a redução de tráfego que proporciona (e consequente redução de congestão e colisões), tem um impacto muito maior na taxa de entrega.

Para tempos de pausa mais elevados, a taxa de entrega é maior para os quatro algoritmos, devido sobretudo à redução de ocorrência de quebras de rota consequente da redução de tráfego imposta e pela diminuição da movimentação na rede.

Comparando o PAMPA com o PAMPA2, as taxas de entrega são muito semelhantes. Para os tempos de pausa menores, 0 e 100 segundos, isto é, para cenários com muito movimento, o PAMPA tem uma taxa de entrega ligeiramente mais reduzida do que o PAMPA2, já que é mais susceptível a quebras de rota. Porém, para tempos de pausa maiores, 300 e 500 segundos, que implicam menos movimento, o PAMPA apresenta uma maior taxa de entrega.

O aumento da estabilidade nas rotas do AODV+P, consequente da diminuição do movimento, aliado ao facto de possuir sempre rotas mais curtas, pode justificar a maior taxa de entrega. Ou seja, menos saltos nas rotas, implica menos retransmissões na rede e, consequentemente, menos ocorrência de colisões e atenuação dos visíveis problemas de congestão.

Por outro lado, o PAMPA2 não selecciona os dispositivos mais afastados do emissor, para retransmitir. Portanto, é possível que uma mensagem de pedido de rota não seja entregue a um destinatário que se localize na extremidade da área de simulação. Ou seja, para cada retransmissão, os dispositivos mais afastados do emissor cancelam as suas retransmissões e, consequentemente, a cobertura proporcionada pelos retransmissores pode ser insuficiente para incluir algum destinatário localizado na extremidade da área

de simulação. A pouca movimentação verificada nos cenários de tempo de pausa $300s$ e $500s$, neste tipo de situações, pode levar a que nunca seja encontrada uma rota para um destinatário. O PAMPA não apresenta esta desvantagem, pois selecciona os dispositivos mais afastados do emissor para retransmitir. Este factor, também pode justificar a maior taxa de entrega do AODV+P para os cenários de menos movimento.

4.7 Sumário

É seguro assumir que o AODV, para todas as versões estudadas neste trabalho, não distribui o tráfego, com equidade, por todos os participantes da rede, em termos de retransmissões de pedidos de rota e de mensagens de dados. Esta distribuição desigual, reflecte-se em problemas de congestão, quando se verifica muito tráfego na rede. Note-se que os testes realizados, simularam uma carga elevada no tráfego da rede.

Os resultados obtidos confirmam a redução que era esperada no tráfego de pedidos de rota do AODV, quando lhe é aplicado o PAMPA e o PAMPA2. O AODV+P concretiza até menos 72% de pedidos de rota do que o AODV e 55% do que o AODV+G. O AODV+P2 reduz o tráfego de pedidos de rota até 74% relativamente ao AODV e até 59% relativamente ao AODV+G.

As rotas utilizadas pelo AODV+P são significativamente mais curtas do que as rotas utilizadas pelo AODV+P2, AODV+G e AODV, obrigando a mais operações de descoberta de rota em cenários de muito movimento. Ou seja, as rotas obtidas pelo AODV+P são mais vulneráveis a quebras do que as do AODV+P2, levando a mais operações de descoberta de rota. Isto justifica a ligeira superioridade na redução de tráfego de pedidos de rota do AODV+P2, relativamente ao AODV+P.

As condições de rede menos favoráveis, reflectiram-se na estabilidade das rotas. Foram realizados bastante mais pedidos de rota do que se esperaria. No entanto, as reduções de tráfego de pedidos de rota, impostas pelo PAMPA e pelo PAMPA2, reflectiram-se, positivamente, na quantidade de números de pedidos de rota efectuados. O AODV+P2 realiza sempre menos pedidos de rota do que qualquer um dos outros algoritmos, porque impõe a maior redução de tráfego. A vulnerabilidade das rotas do AODV+P, justifica porque esta versão do protocolo realiza sempre mais pedidos de rota do que o AODV+P2. Justifica também, porque o AODV+P realiza mais pedidos de rota, nos cenários de tempo de pausa 0, do que o AODV+G e o AODV.

O AODV+P e o AODV+P2 usufruem de menos latência na entrega de mensagens de dados e nas operações de descoberta de rota, relativamente ao AODV+G e ao AODV. A redução de tráfego de proporcionado pelo PAMPA e PAMPA2, atenuaram os problemas de congestão, colisões e contenção, justificando esta vantagem do AODV+P e do AODV+P2. O AODV+P2 sofre de mais latência do que o AODV+P, pois impõe valores de atraso maiores aos dispositivos que não se encontram na distância favorecida pela

função.

As mesmas vantagens de redução de tráfego, proporcionadas pelo PAMPA e pelo PAMPA2, garantem ao AODV+P e ao AODV+P2, maior taxa de entrega de mensagens de dados, relativamente ao AODV+G e ao AODV. O AODV+P e o AODV+P2 concedem ganhos na taxa de entrega, até 9% relativamente ao AODV+G e até 15% relativamente ao AODV. A vulnerabilidade das rotas do AODV+P, faz com que a sua taxa de entrega seja menor do que a do AODV+P2. Porém, o facto de o PAMPA proporcionar rotas mais curtas, favorece o AODV+P em termos de taxa de entrega, nos cenários de pouca movimentação.

Capítulo 5

Implementação

A implementação do PAMPA num ambiente não simulado, permite o desenvolvimento de aplicações para MANETs baseadas neste algoritmo de difusão e vai permitir, futuramente, avaliar o impacto das condições reais, no desempenho do algoritmo. Este capítulo descreve uma concretização do PAMPA que permitiu confirmar, bem como avaliar, os requisitos mínimos de execução.

A capacidade dos participantes de uma MANET, de obter a intensidade da força de sinal com que receberam uma mensagem, é fundamental para o PAMPA, pois vai permitir o cálculo da atraso a ser imposto na retransmissão de uma mensagem recebida. Porém, este aspecto está condicionado pelo suporte fornecido pelos controladores do dispositivo de rede. O suporte para a obtenção da intensidade da força de sinal relativa a um dispositivo de uma rede ad hoc, é muitas vezes abandonado, por lhe ser atribuída pouca ou nenhuma utilidade.

5.1 Ambiente de desenvolvimento

Algumas operações do algoritmo, como a obtenção da intensidade da força de sinal, necessitam de uma chamada ao sistema operativo, que por sua vez, realiza a respectiva chamada ao controlador do dispositivo.

Assim, diferentes sistemas operativos, requerem diferentes implementações. Esta implementação do PAMPA foi concretizada em ambiente Linux, por razões que se prendem, sobretudo, com a conhecida característica deste sistema operativo ser baseado em software aberto, ou seja, o código fonte do sistema, das aplicações e dos controladores, é disponibilizado.

A biblioteca wireless do Linux, denominada por *iwlib*, proporciona operações para aplicações de redes sem fios, invocando funções do sistema operativo que, por sua vez, comunicam directamente com os controladores do dispositivo de rede. Esta biblioteca possui funções que permitem manipular os dispositivos de rede, invocando as funções apropriadas do sistema. Na Listagem 5.1 é apresentado o código fonte de uma função,

pertencente à biblioteca *iwlib.h*, que permite a execução de operações nos controladores, verificando-se que requer quatro parâmetros, a que a seguir nos referimos:

- O dispositivo necessita de uma *socket* para executar as operações, sendo o descritor de ficheiro passado no primeiro argumento da função, cujo nome é *skfd*.
- O parâmetro *ifname*, que representa o nome da interface de rede.
- O parâmetro *request*, que especifica a operação que se pretende que o dispositivo execute. Estão definidas *flags* correspondentes a cada uma das operações disponibilizadas pelo dispositivo, podendo ser passadas por este parâmetro, indicando qual a operação que se pretende.
- A estrutura de nome *pwrq*, passada por parâmetro, inclui a informação requerida para a operação a ser executada, bem como campos onde vão ser colocados os resultados.

```
1
  static inline int
3 iw_get_ext(int       skfd,    /* Socket to the kernel */
       const char *   ifname,    /* Device name */
5      int       request,  /* WE ID */
       struct iwreq * pwrq)    /* Fixed part of the request */
7 {
    /* Set device name */
9   strncpy(pwrq->ifr_name, ifname, IFNAMSIZ);

11  /* Do the request */
    return(ioctl(skfd, request, pwrq));
13 }
```

Listing 5.1: iwlib.h

Pode-se verificar que esta função invoca outra de nome *ioctl* que pertence a uma biblioteca do sistema.

A *flag SIOCGIWSPY* representa um pedido que permite obter todos os endereços físicos, bem como as correspondentes intensidades da força de sinal de todos os dispositivos vizinhos. Assim, obter a intensidade da força de sinal de um determinado participante vizinho, passa por invocar a função apresentada na listagem 5.1, passando a *flag SIOCGIWSPY* no parâmetro *request*. Porém, se os controladores do dispositivo de rede não tiverem suporte para esta operação, a função *iw_get_ext* não vai preencher a estrutura *pwrq* com os valores de intensidade de força de sinal, e retorna -1.

Muitos controladores para Linux não proporcionam a operação pretendida. Em consequência, para concretizar uma implementação do PAMPA para ambiente Linux, foi necessário procurar controladores de dispositivos de redes que proporcionassem o suporte. Foram então, encontrados controladores em [2] denominados *MadWifi*. Estes controladores foram desenvolvidos para placas de rede com o *chipset* da marca *Atheros* [1], em

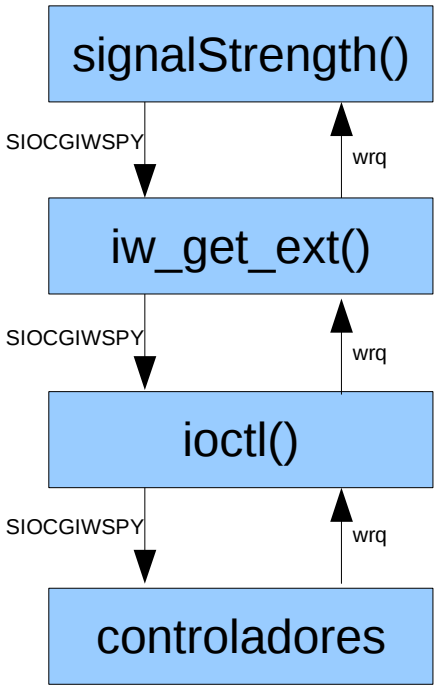


Figura 5.1: Pedido aos controladores

ambiente Linux. Isto permitiu o desenvolvimento de uma função capaz de obter o valor da intensidade da força de sinal relativa a um dispositivo vizinho de uma rede ad hoc, através do respectivo endereço físico. Porém, pretendia-se a obtenção da intensidade da força de sinal relativa a uma mensagem recebida, mas não foram encontrados controladores que a permitissem obter desta forma. No entanto, a função apresentada na listagem 5.2, pode ser invocada cada vez que uma mensagem é recebida, passando o endereço físico do emissor no parâmetro *senderAddr*. Uma noção básica do processo de obtenção do valor da intensidade da força de sinal, está ilustrado na Figura 5.1.

```
static int
signalStrength(int  skfd,char * ifname,
  char *  args[],int  count,char * senderAddr) {

  struct iwreq    wrq;
  char    buffer[(sizeof(struct iw_quality) +
      sizeof(struct sockaddr)) * IW_MAX_SPY];
  char    temp[128];
  struct sockaddr * hwa;
  struct iw_quality * qual;
  iwrange range;
  int   has_range = 0;
  int   n;
  int   i;
```

```
16
    /* Collect stats */
18  wrq.u.data.pointer = (caddr_t) buffer;
    wrq.u.data.length = IW_MAX_SPY;
20  wrq.u.data.flags = 0;

22  if(iw_get_ext(skfd, ifname, SIOCGIWSPY, &wrq) < 0)
     {
24      return(PAMPA_FAILED_SIGNAL_STRENGTH);
      }
26
    /* Number of addresses */
28  n = wrq.u.data.length;

30  /* Check if we have valid mac address type */
    if(iw_check_mac_addr_type(skfd, ifname) < 0)
32   {
        return(PAMPA_FAILED_SIGNAL_STRENGTH);
34     }

36  /* Get range info if we can */
    if(iw_get_range_info(skfd, ifname, &(range)) >= 0)
38    has_range = 1;

40  /* The two lists */
    hwa = (struct sockaddr *) buffer;
42  qual = (struct iw_quality *) (buffer + (sizeof(struct sockaddr) * n))
       ;

44  for(i = 0; i < n; i++) {
        /* return stats */
46      if(strcmp(iw_saether_ntop(&hwa[i], temp),senderAddr)==0){
          senderQual = (float)qual[i].qual;
48        maxQual = (float)range.max_qual.qual;

50        return (int) 1000000 * (senderQual/maxQual);
        }
52    }
    return(PAMPA_FAILED_SIGNAL_STRENGTH);
54 }
```

Listing 5.2: Função desenvolvida

Podemos verificar na listagem 5.2, na linha 22, que a função desenvolvida, denominada *signalStrength*, invoca a função apresentada na listagem 5.1 com o pedido *SIOCGIWSPY*. Se os controladores do dispositivo não suportarem a operação, a função retorna o valor de erro definido por *PAMPA_FAILED_SIGNAL_STRENGTH*. Caso contrário, os endereços físicos e os respectivos valores de intensidade de força de sinal são colocados num campo da estrutura *wrq*. Na linha 46 a tabela de endereços físicos (dada por *hwa[]*, da estrutura *wrq*, é percorrida até ser encontrado o endereço físico, especificado no parâmetro *senderAddr* da função *signalStrength* e o valor de intensidade de força de sinal, presente na tabela *qual[]*, é retornado.

A partir desta função, foi desenvolvida uma biblioteca na linguagem C, que permite implementar o PAMPA. Porém, como assenta na função visualizada na Figura 5.2, requer que os controladores do dispositivo de rede, suportem a operação *SIOCGIWSPY*.

5.2 Interface da biblioteca

À biblioteca desenvolvida dá-se o nome *lib_pampa.h* e as suas funções estão apresentadas na listagem 5.3.

```
2
  /*Initiates PAMPA, opens all the necessary sockets,
4  *       allocates the required memory
   *                initiates the necessary variables and threads.
6  *
   *@param ifname     The network interface to be used (example, "ath0")
8  *@param user     The user ip address and port
   *@param kons     The PAMPA pre-defined constant
10 *@param nRetransmitions  The PAMPA pre-defined treshold
   *
12 *@return   The structure containing all the necessary information to
       use PAMPA
   *@return NULL  If unable to initiate all the necessary operations
14 */
  struct pampa_context * openPAMPA(char * ifname, struct sockaddr_in *
     user,
16          int kons,int nRetransmitions);

18 /*Broadcasts a message
   *
20 *@param ctx      The structure containing all the necessary
       information to use PAMPA
   *@param receivedMessage  The message received
22 *@param flags      Don't wait for the next message: 1; Other int value
   *@param from     The sender's address
24 *@param maxBufferSize    The maximum number of bytes to be received
   *
26 *@return   The number of bytes received
   *@return -1  If no message was received
28 *@return -1    If error
   */
30 int recvPAMPA(struct pampa_context * ctx,char * receivedMessage,
      int flags, struct sockaddr_in * from,int maxBufferSize);
32
  /*Broadcasts a message
34 *
   *@param ctx    The structure containing all the necessary information
       to use PAMPA
36 *@param msg    The message to be sent
   *@param sizeMsg  The size of the message to be sent
38 *
   *@return   The number of bytes successfully sent
```

```
40  *@return -1  If fails
    */
42 int sendPAMPA(struct pampa_context * ctx,char * msg, int sizeMsg);

44 /*Terminates PAMPA, frees de allocated memory
    *
46  *@param ctx  The structure containing all the necessary information to
        use PAMPA
    *
48  *@return 0   if success
    *@return -1  otherwise
50  */PAMPA_NO_BLOCK
   int closePAMPA(struct pampa_context * ctx);
```

Listing 5.3: lib_pampa.h

openPAMPA A função *openPAMPA* vai inicializar, dentro de uma estrutura do tipo *struct pampa_context*, todas as *sockets*, *threads*, e variáveis, necessárias ao funcionamento do PAMPA. Esta estrutura é retornada pela função e é fundamental para a execução das restantes operações da biblioteca. A função requer o nome da interface do dispositivo, a ser passada pelo parâmetro *ifname*, não só porque se torna necessário para a operação de obtenção da intensidade da força de sinal, visualizada na listagem 5.1, mas também para verificar se os controladores suportam a operação. O parâmetro *kons* permite especificar o valor da constante pré-definida do PAMPA, sendo preferível que todos os dispositivos pertencentes à MANET, atribuam o mesmo valor a este parâmetro. O número de cópias a receber de uma mensagem, para que a respectiva retransmissão agendada seja cancelada, é colocado no parâmetro *nRetransmitions*. Pode ser atribuído o valor 0 aos dois últimos parâmetros referidos, com o intuito de usar os valores por defeito. O valor por defeito de *kons* é 5, e o de *nRetransmitions* é 1. O parâmetro *user* contém o endereço e o porto a utilizar pelo dispositivo. É inicializada uma *thread* de execução para a recepção de mensagens dos outros participantes da MANET.

recvPAMPA Os conteúdos das mensagens recebidas pela primeira vez são guardados numa fila, dentro da estrutura devolvida pela função *openPAMPA* e podem ser obtidos através da função *recvPAMPA*. A estrutura retornada pela função *openPAMPA* é recebida como parâmetro, sendo assim possível que a função *recvPAMPA*, aceda aos conteúdos das mensagens. O conteúdo que se encontra no início da fila, é colocado no parâmetro *receivedMessage*, o endereço do seu emissor é colocado no parâmetro *from* e o tamanho é retornado pela função. Se a fila estiver vazia, a função bloqueia até um elemento ser recebido, a menos que a *flag PAMPA_NO_BLOCK* seja passada no argumento *flags*, fazendo com que a função retorne o valor -1, caso a fila esteja vazia. O parâmetro *maxBufferSize* impõe um valor máximo de bytes a serem recebidos, sendo que os recebidos em excesso, são descartados.

sendPAMPA A função *sendPAMPA* difunde a mensagem colocada no parâmetro *msg*, de tamanho especificado no parâmetro *sizeMsg*.

closePAMPA A função *closePAMPA* liberta toda a memória alocada para o funcionamento do PAMPA e fecha todas as sockets e semáforos.

struct pampa_context Como se pode verificar na listagem 5.3, a invocação das funções *recvPAMPA*, *sendPAMPA* e *closePAMPA* está dependente da estrutura retornada pela função *openPAMPA*. Os campos da função podem ser visualizados na listagem 5.4. O valor da constante e do número máximo de retransmissões, são guardados nas primeiras duas variáveis. O tamanho máximo das mensagens é guardado na variável *maxSize*. A variável *currentID* é um contador que permite que os receptores identifiquem as mensagens enviadas e, as variáveis headID e tailID correspondem à cabeça e cauda da lista ligada dos identificadores das mensagens recebidas. As mensagens enviadas são identificadas pelo par $< endereçoIP, currentID >$, e são armazenadas nesta lista, até à invocação da função *closePAMPA*. O conteúdo das mensagens não é armazenado, para não sobrecarregar a memória. São criadas as sockets, *recvSocket* e *brdcstSocket*, para receber mensagens e difundir mensagens, respectivamente. A variável *ifname* contém o nome da interface, a variável *mac* o endereço físico, necessárias para invocar a função *signalStrength*. É criada uma lista ligada de *threads*, cuja cabeça é a variável *head*, para lidar com as retransmissões de mensagens, sabendo que podem ser recebidas várias em simultâneo. O conteúdo das mensagens recebidas, é colocado numa lista ligada, onde as variáveis *head_msg_queue* e *tail_msg_queue* correspondem à sua cabeça e cauda respectivamente. Algumas variáveis presentes na listagem, não foram referidas, por terem pouca relevância para a apresentação da biblioteca.

```
1
  typedef struct pampa_context
3 {

5   /*this constant will allow us to calculate the waiting time
      before broadcasting the message*/
7   double kons;/*the contant*/

9   int numRtrsmt;/*number of retransmitions allowed for each message*/

11  int maxSize; /*maximum size of the messages*/

13  unsigned int currentID;/*allows the messages we send to be identified
        uniquely*/

15  /*the message identifiers of the received messages
      are stored in a linked list*/
17  struct message_id * headID;/*head of the linked list*/
    struct message_id * tailID;/*tail of the linked list*/
```

```
19   int numIds;/*size of the linked list*/

21   /*all the sockets we need*/
     int recvSocket;/*socket to receive messages*/
23   int brdcstSocket;/*socket to broadcast messages*/

25   char * ifname;/*device name*/
     char * mac;
27   int recvPort;/*my port to receive messages*/
     int brdcstPort;/*my port to broadcast messages*/
29

31   struct sockaddr_in in;/*my address for recvSocket*/
     struct sockaddr_in me;/*my address for brdcstSocket*/
33   struct sockaddr_in out;/*destination address for brdcstSocket*/

35   struct thread_node * head;/*head of the thread linked list*/
     int numThreads;/*the number of threads created until now -> size of
        the list*/
37
     struct receive_context * recvContext;/*receive stuff*/
39
     msg_queue * head_msg_queue; /*head of the message queue*/
41   msg_queue * tail_msg_queue; /*tail of the message queue*/

43   sem_t * sem_msg_queue_full;
     sem_t * sem_msg_queue_empty;
45
   } pampa_ctx;
```

Listing 5.4: struct pampa_context

As *flags* definidas para esta biblioteca, são da maior relevância, e estão apresentadas na listagem 5.5.

```
1
   /*errors and stuff*/
3  #define PAMPA_FAIL_CLOSE_SOCKET -13
   #define PAMPA_FAIL_CLOSE_SEMAPHORE -12
5  #define PAMPA_FAIL_SEMAPHORE -11
   #define PAMPA_DRIVER_NOT_SUPPORTED -10 /*the driver does not support
      iwspy*/
7  #define PAMPA_FAILED_SIGNAL_STRENGTH -9 /*unable to obtain de signal
      strength*/
   #define PAMPA_EMPTY_LIST -8 /*The message linked list is empty*/
9  #define PAMPA_FAILED_MAC -7 /*Unable to obtain the mac address*/
   #define PAMPA_NEW_MSG -6 /*received a new message :) */
11 #define PAMPA_FAILED_SEND -5 /*failed to send the message*/
   #define PAMPA_FAILED_BIND -4 /*failed to bind a socket*/
13 #define PAMPA_FAILED_SOCKET -3 /*failed to open a socket*/
   #define PAMPA_FAILED_THREAD -2 /*failed to start a thread*/
15 #define PAMPA_MSG_DISCARD -1 /*the message received is to be discarded
      */
   #define PAMPA_SUCCESS 0;
17
```

```
#define PAMPA_NO_BLOCK 1;
19


21
#define PAMPA_DFT_KONS 5; /*Default PAMPA constant*/
23 #define PAMPA_DFT_RETRANSM 1; /*Default PAMPA treshold*/
```

Listing 5.5: flags

Foram, portanto, criadas as condições necessárias para a implementação do PAMPA em aplicações de rede, para Linux.

5.3 Experimentação

Recorrendo à biblioteca *lib_pampa*, foi desenvolvida uma simples aplicação de difusão de pequenas mensagens de texto, entre participantes de uma rede ad hoc, a ser executada numa *shell* Linux. A aplicação tem início com a invocação da função *openPAMPA* e termina com a execução da função *closePAMPA*. O utilizador escreve uma mensagem na consola, sendo esta difundida através da função *sendPAMPA*. As mensagens são recebidas pelas aplicações dos participantes, através da função *recvPAMPA*, que se encontra bloqueada a aguardar por mensagens, dentro de uma *thread*.

Para testar a aplicação, foram utilizados dois computadores portáteis e um fixo. Todos tinham integrada uma placa de rede com o *chipset* da marca *Atheros* sendo, por isso, utilizados controladores *MadWifi*, que suportam a operação *SIOCGIWSPY*. O sistema operativo utilizado, foi o *Linux Knoppix 5.3*.

Note-se que não foi garantido que as três placas de rede tivessem o mesmo raio de alcance. No entanto, isto não afecta a implementação do PAMPA, nem esta experimentação, pois um dispositivo mais afastado do emissor, em princípio, obtém um valor de força de sinal mais baixo.

Os computadores foram colocados em modo ad hoc e associados à mesma rede. De seguida, foi iniciada a aplicação em cada um deles. As funções da biblioteca *lib_pampa* não imprimem qualquer tipo de mensagem para a consola e consequentemente, não é possível observar os detalhes do funcionamento do PAMPA. Assim, para este teste, foram adicionadas três impressões, através da função *printf*, ao código fonte da biblioteca, para apurar o tempo de espera imposto a uma mensagem recebida pela primeira vez e notificar o utilizador se uma mensagem foi cancelada ou se foi retransmitida.

Foi atribuído o valor 0.5 à constante do PAMPA e o valor 1 à variável de *treshold* do PAMPA. No início da aplicação é especificando o nome da interface, endereço, porto, constante do PAMPA e *treshold*.

Um dos computadores portáteis foi seleccionado para difundir mensagens, uma de cada vez, observando-se a reacção dos outros dois. A planta do local onde foi realizada esta experimentação, pode ser visualizada na Figura 5.2.

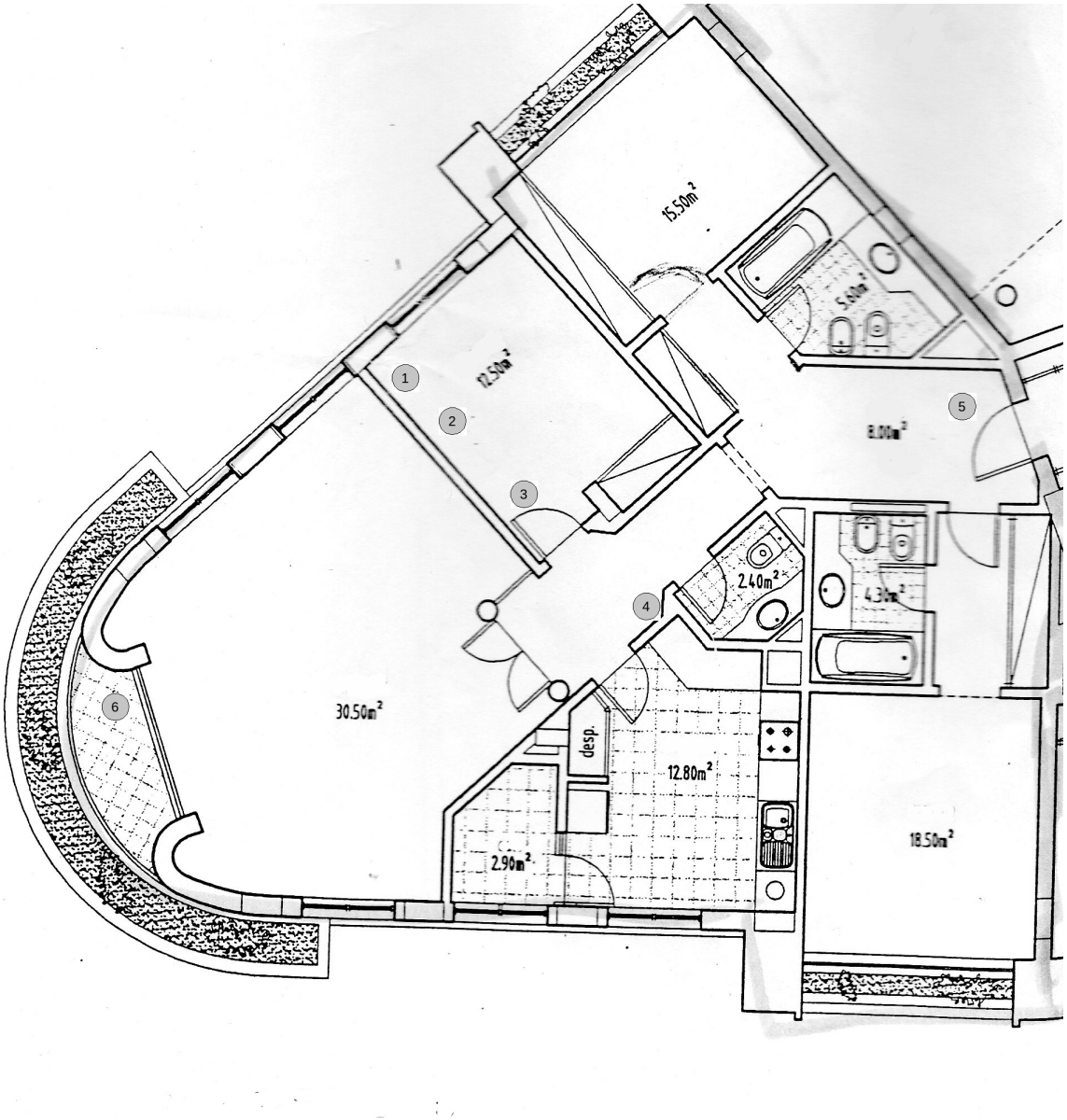


Figura 5.2: Planta

Os seis locais, onde os computadores foram colocados, estão identificados na planta apresentada. O computador fixo, encontra-se sempre na posição marcada com o número 2. Os dois computadores portáteis foram colocados nas restantes posições para os diferentes testes efectuados.

Foi atribuído o endereço IP $192.168.1.1$ ao dispositivo portátil emissor. Ao outro computador portátil, foi atribuído o endereço $192.168.1.2$, e ao computador fixo o endereço $192.168.1.3$. Os endereços são impressos durante os testes, de modo a auxiliar a interpretação das figuras.

Primeiro teste Para o primeiro envio, o computador portátil emissor foi colocado no local marcado com o número 4 e o segundo receptor no número 3. As duas posições estão separadas por $4m$, sendo que a posição 3 distância-se $2m$ do computador fixo (posição 2). Na figura 5.3(a), pode ser visualizado o início do teste, que consiste na difusão da mensagem "teste1", a partir da posição 4.

É de salientar que a aplicação termina se o utilizador inserir 1, sendo invocada a função *closePAMPA*. Ao ser inserido 2, o utilizador deve escrever uma mensagem, que será difundida através da função *sendPAMPA*

```
knoppix@Microknoppix:~/Desktop$ sudo ./run wlan0 192.168.1.1 4567 0.5 1
starting pampa....
done! -> insert 1 to terminate pampa, insert 2 to broadcast a message
>2
write message (maximum 100 characters): teste1
>
```

(a) Difusão da mensagem

```
knoppix@Knoppix:/ramdisk/home/knoppix/Desktop/runPampa$ sudo ./run ath0 192.168.1.2 4567 0.5 1
starting pampa....
done! -> insert 1 to terminate pampa, insert 2 to broadcast a message
>mensagem recebida de 192.168.1.1: teste1
mensagem 1 em fila; tempo de espera: 0.485289 segundos
A retransmissão da mensagem 1 foi cancelada
```

(b) Cancelamento da retransmissão

```
knoppix@Knoppix:/ramdisk/home/knoppix/Desktop/runPampa$ sudo ./run ath0 192.168.1.3 4567 0.5 1
starting pampa....
done! -> insert 1 to terminate pampa, insert 2 to broadcast a message
>mensagem 1 em fila; tempo de espera: 0.442730 segundos
mensagem recebida de 192.168.1.1: teste1
A mensagem 1 foi retransmitida
```

(c) Retransmissão

Figura 5.3: Primeiro teste

Nas figuras 5.3(b) e 5.3(c) podemos visualizar a recepção da mensagem nos dispositivos localizados nas posições 3 e 2, respectivamente. Pode-se verificar que, o computador

fixo calcula um intervalo de tempo de espera de $0.442730s$ e retransmite, obrigando o outro dispositivo a cancelar a sua retransmissão, sendo que calculou um intervalo de tempo de espera de $0.485289s$. Note-se que, os dois receptores estavam distanciados por $2m$ e que a diferença entre os dois intervalos de tempo calculados, era de apenas 4 centésimas de segundo, aproximadamente.

Segundo Teste Para a segunda parte da experimentação, colocou-se o computador portátil na posição 1 da Figura 5.2, mantendo-se os outros dois nas mesmas posições. Portanto, o dispositivo $192.168.1.2$, que no teste anterior localizava-se mais próximo do emissor, passou a ser o mais afastado. A posição 1 e a posição 2 são separadas por $1m$, logo a posição 2 está a $6m$ do emissor e a posição 3 a $7m$.

```
knoppix@Microknoppix:~/Desktop$ sudo ./run wlan0 192.168.1.1 4567 0.5 1
starting pampa....
done! -> insert 1 to terminate pampa, insert 2 to broadcast a message
>2
write message (maximum 100 characters): teste1
>2
write message (maximum 100 characters): teste2
>
```

(a) Difusão da mensagem

```
knoppix@Knoppix:/ramdisk/home/knoppix/Desktop/runPampa$ sudo ./run ath0 192.168.1.2 4567 0.5 1
starting pampa....
done! -> insert 1 to terminate pampa, insert 2 to broadcast a message
>mensagem recebida de 192.168.1.1: teste1
mensagem 1 em fila; tempo de espera: 0.485289 segundos
A retransmissão da mensagem 1 foi cancelada
mensagem recebida de 192.168.1.1: teste2
mensagem 2 em fila; tempo de espera: 0.371207 segundos
A mensagem 2 foi retransmitida
```

(b) Retransmissão

```
knoppix@Knoppix:/ramdisk/home/knoppix/Desktop/runPampa$ sudo ./run ath0 192.168.1.3 4567 0.5 1
starting pampa....
done! -> insert 1 to terminate pampa, insert 2 to broadcast a message
>mensagem 1 em fila; tempo de espera: 0.442730 segundos
mensagem recebida de 192.168.1.1: teste1
A mensagem 1 foi retransmitida
mensagem recebida de 192.168.1.1: teste2
mensagem 2 em fila; tempo de espera: 0.428332 segundos
A retransmissão da mensagem 2 foi cancelada
```

(c) Cancelamento da retransmissão

Figura 5.4: Segundo teste

O decorrer do teste pode ser visualizado nas figuras apresentas em 5.4. O teste anterior ainda pode ser visualizado nestas figuras, sendo que no teste actual foi enviada a mensagem *teste2*. Podemos verificar na Figura 5.4(b), que foi o dispositivo portátil, loca-

lizado na posição 1, a retransmitir, o que era esperado, sabendo que se encontrava mais afastado do emissor do que o computador fixo localizado na posição 2. O cancelamento da mensagem por parte do computador fixo, é visível na Figura 5.4(c).

Terceiro Teste Para o terceiro e último teste, o dispositivo emissor foi colocado na posição 5 e o outro portátil na posição 6. Desta forma garantiu-se que, o dispositivo portátil receptor localizava-se fora do raio de alcance do emissor, mas dentro do raio de alcance do computador fixo (ainda na posição 2). Este, por sua vez, localizava-se dentro do alcance do emissor.

Portanto, pretendia-se que, a mensagem fosse transmitida na posição 5, retransmitida na posição 2 e recebida na posição 6. O decorrer do teste está ilustrado na Figura 5.5. Note-se que a ordem das figuras apresentadas em 5.5 mudou, relativamente às figuras apresentadas em 5.3 e 5.4. A reacção do dispositivo portátil receptor, é apresentada na Figura 5.5(c) e a do computador fixo na Figura 5.5(b).

```
knoppix@Microknoppix:~/Desktop$ sudo ./run wlan0 192.168.1.1 4567 0.5 1
starting pampa....
done! -> insert 1 to terminate pampa, insert 2 to broadcast a message
>2
write message (maximum 100 characters): teste3
>
```

(a) Transmissão da mensagem

```
knoppix@Knoppix:/ramdisk/home/knoppix/Desktop/runPampa$ sudo ./run ath0 192.168.1.3 4567 0.5 1
starting pampa....
done! -> insert 1 to terminate pampa, insert 2 to broadcast a message
>mensagem 1 em fila; tempo de espera: 0.142721 segundos
mensagem recebida de 192.168.1.1: teste3
A mensagem 1 foi retransmitida
```

(b) Retransmissão

```
knoppix@Knoppix:/ramdisk/home/knoppix/Desktop/runPampa$ sudo ./run ath0 192.168.1.2 4567 0.5 1
starting pampa....
done! -> insert 1 to terminate pampa, insert 2 to broadcast a message
>mensagem 1 em fila; tempo de espera: 0.099789 segundos
mensagem recebida de 192.168.1.3: teste3
A mensagem 1 foi retransmitida
```

(c) Recepção

Figura 5.5: Terceiro teste

Pode-se verificar na Figura 5.5(b), que a mensagem é recebida, do emissor, pelo computador fixo, de endereço 192.168.1.3. Note-se que o endereço IP do emissor, 192.168.1.1, é apresentado na recepção da mensagem, pela aplicação do computador fixo. Esta é retransmitida, e recebida pelo computador portátil, na posição 6, de endereço 192.168.1.2.

O endereço IP do computador fixo é apresentado pela aplicação na recepção da mensagem. Portanto, o teste decorreu como pretendido.

5.4 Sumário

Foi concretizada uma biblioteca, na linguagem C e em ambiente Linux, que permite implementar o PAMPA em aplicações de rede. Porém, é necessário que os controladores da placa de rede, suportem a operação *SIOCGIWSPY*, para que o seja possível realizar a operação de obtenção de força de sinal. Esta implementação não permite obter a intensidade da força de sinal relativa a uma mensagem recebida. No entanto, tem a capacidade de obter a intensidade da força de sinal de todos os dispositivos dentro do seu raio de alcance. A ideia passa por aplicar esta funcionalidade, cada vez que uma mensagem é recebida.

A biblioteca foi testada com três computadores, em ambiente Linux. Mostrou ser capaz de cancelar retransmissões, quando o retransmissor se encontra ligeiramente mais afastado. Os testes revelam que a biblioteca respeita todos os aspectos do algoritmo.

O impacto das condições reais no desempenho do PAMPA, é uma temática da maior relevância, que terá de ser estudada futuramente.

Capítulo 6

Conclusão

Dos estudos e testes efectuados para a elaboração deste trabalho, concluiu-se que o número excessivo de retransmissões numa MANET acarreta consequências severas, ao conduzir a um consumo desnecessário de recursos computacionais, comunicacionais e energéticos da MANET, bem como a uma perda de largura de banda.

O flooding é um algoritmo de difusão que visa entregar cada mensagem ao maior número possível de participantes numa MANET, porém com um custo muito elevado de retransmissões, contribuindo, assim, para a ocorrência do problema de *Broadcast Storms*. De facto, os protocolos de encaminhamento que utilizam o flooding, como o AODV, sobrecarregam a rede com retransmissões excessivas comprometendo, consideravelmente, o seu desempenho.

Algoritmos probabilistas, como o GOSSIP3(p, k, m), contribuem para uma atenuação deste problema, reduzindo o número de retransmissões, ainda que com limitações na adaptação dinâmica a diferentes condições de rede, por não possuir qualquer critério na selecção dos dispositivos que vão retransmitir. Este facto, pode não garantir a entrega de mensagens a todos os participantes da rede.

Foram analisadas as vantagens da aplicação do algoritmo de difusão PAMPA a protocolos de encaminhamento, nomeadamente o AODV. Os resultados obtidos, relativamente às simulações realizadas, demonstram que as características únicas do PAMPA contribuem para uma redução de tráfego bastante mais significativa, sendo certo que, assim, melhoram consideravelmente o desempenho do protocolo. Porém, estas características tornam o protocolo menos tolerante ao movimento dos participantes, em consequência da selecção dos participantes mais afastados do emissor. Ainda assim, com este trabalho, provou-se que o PAMPA é uma opção melhor e mais viável, para protocolos de encaminhamento, do que o flooding e o GOSSIP3(p, k, m).

Foi igualmente apresentado um novo algoritmo de difusão baseado em distância, em tudo semelhante ao PAMPA, que proporciona mais tolerância ao movimento dos dispositivos na MANET, denominado PAMPA2. Para cada transmissão e retransmissão, o PAMPA2 selecciona participantes localizados numa zona intermédia do raio de alcance do

emissor, facto que o torna menos susceptível a quebras de rota. O PAMPA2 oferece melhor desempenho no protocolo, comparativamente ao PAMPA, em MANETs com maior movimento dos participantes.

Foi também criada uma biblioteca na linguagem C, que permite implementar PAMPA em qualquer distribuição Linux, para placas de rede IEEE 802.11 (WiFi). No entanto, é necessário que os controladores do dispositivo de rede suportem a operação identificada por *SIOCGIWSPY*.

Os resultados da execução de uma aplicação de testes em ambiente real sugerem que a biblioteca apresentada, concretiza todas as características do PAMPA.

Bibliografia

[1] Atheros communications: http://www.atheros.com/.

[2] The madwifi project: http://madwifi-project.org/.

[3] Network simulator 2: http://www.isi.edu/nsnam/ns/.

[4] Christian Bettstetter, Giovanni Resta, and Paolo Santi. The node distribution of the random waypoint mobility model for wireless ad hoc networks. *IEEE Trans. on Mobile Computing*, 2(3):257–269, 2003.

[5] Laura Marie Feeney and Martin Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *Procs. of the 20th Joint Conf. of the IEEE Comp. and Comm. Societies (INFOCOM 2001)*, volume 3, pages 1548–1557, 2001.

[6] Zygmunt J. Haas, Joseph Y. Halpern, and Li Li. Gossip-based ad hoc routing. In *Procs. of the 21st Joint Conf. of the IEEE Comp. and Comm. Societies (INFOCOM 2002)*, volume 3, pages 1707–1716, 2002.

[7] David B. Johnson and David A. Maltz. *Mobile Computing*, chapter Dynamic Source Routing in Ad Hoc Wireless Networks, pages 153–181. Kluwer Academic Publishers, 1996.

[8] David B. Johnson, David A. Maltz, and Josh Broch. *Ad Hoc Networking*, chapter DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks, pages 139–172. Addison-Wesley, 2001.

[9] Hugo Miranda, Simone Leggio, Luís Rodrigues, and Kimmo Raatikainen. A power-aware broadcasting algorithm. In *Procs. of The 17th IEEE Int'l Symposium on Personal, Indoor and Mobile Radio Comm. (PIMRC'06)*, 2006.

[10] Hugo Miranda, Simone Leggio, Luís Rodrigues, and Kimmo Raatikainen. Removing probabilities to improve efficiency in broadcast algorithms. In *Procs. of the 5th MiNEMA Works.*, 2007.

[11] Charles E. Perkins and Elizabeth M. Royer. Ad-hoc on-demand distance vector routing. In *Procs. of the 2nd IEEE Works. on Mobile Comp. Systems and Applications*, pages 90–100, 1999.

[12] Theodore S. Rappaport. *Wireless communications: principles and practice*. Prentice Hall, 1996.

[13] Yu-Chee Tseng, Sze-Yao Ni, Yuh-Shyan Chen, and Jang-Ping Sheu. The broadcast storm problem in a mobile ad hoc network. *Wireless Networks*, 8(2/3):153–167, 2002.

[14] Bruce Tuch. Development of wavelan, an ism band wireless lan. *AT&T Technical Journal*, pages 27–37, July/August 1993.

[15] J. Yoon, M. Liu, and B. Noble. Random waypoint considered harmful. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, volume 2, pages 1312–1321 vol.2, March-3 April 2003.